VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Data mining nad znalostmi v e-learningové podpoře pro výuku logiky

Data mining using knowledge on an e-learning support of training logic

2011

Vojtěch PETR

Prohlášení

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě dne 6. 5. 2011 ……

Vojtěch PETR

Poděkování

Chtěl bych tímto poděkovat vedoucímu diplomové práce Mgr. Marku Menšíkovi Ph.D. za pomoc a konzultace. Dále bych touto cestou také rád poděkoval panu Ing. Radoslavu Fasugovi, Ph.D. za jeho ochotu konzultovat řešení problémů.

Abstrakt a klíčová slova

Abstrakt

Cílem této diplomové práce je použití data miningových metod na data získaná e-learningovým systémem eLogika. Hlavním cílem je pak takto získané informace interpretovat uživateli a využít je pro další cílený rozvoj při studiu a výuce.

První část diplomové práce se zabývá obecně data miningem, jsou vysvětleny metodologie vývoje data miningových úloh. V druhé části jsou popsány data miningové metody obsažené v Microsoft SQL Serveru 2008 a jejich použití. V závěrečné části je popsán vývoj a nasazení data miningového projektu v systému eLogika pomocí kroků metodologie CRISP-DM.

Klíčová slova

Data mining, SQL Server 2008, Integration Services, Analysis Services, metodologie CRIPS-DM, Business Intelligence, DMX, e-learning

Abstract and key words

Abstract

The objective of this diploma thesis is to use data mining methods on data obtained by e-learning system eLogika. The main goal is to interpret the received information's to user and using them for targeted improvement of a learning and teaching.

The first part covers general data mining, there are explained methodology's of development of data mining tasks. The second part describes data mining methods contained in Microsoft SQL Server 2008 and their using. The final part describes the development and deployment of data mining project in system eLogika using the CRISP-DM methodology steps.

Key words

Data mining, SQL Server 2008, Integration Services, Analysis Services, methodology CRIPS-DM, Business Intelligence, DMX, e-learning

Seznam použitých symbolů a zkratek

BI	Business intelligence
CRM	Customer relationship management
DMX	Data Mining Extensions
ETL	Extract, transform, load
FK	Foreign Key
LMS	Learning Management System
OLE DB	Object Linking and Embedding for Databases
PK	Private Key
SQL	Standard Query Language
SSIS	SQL Server Integration Services

Obsah

Seznam obrázků

Seznam tabulek

1 Úvod

Data mining se nazývá poměrně mladý obor informatiky, který je momentálně nejrychleji rostoucí segment Business Inteligence. Pomocí data miningu lze z velkých databází vydolovat potenciálně užitečné informace, které mohou uživatelé využít pro další rozhodování. Rozsáhlé možnosti data miningových metod vedly k myšlence využít jejich potenciál v e-learningové výuce, potažmo v systému eLogika.

Hlavním cílem této diplomové práce je aplikace vybraných data miningových metod na data získaná systémem eLogika. Takto získané potenciálně užitečné informace budou uživateli interpretovány a následně využívány pro další cílený rozvoj při studiu a výuce.

V první části této diplomové práce je popsán e-learningový systém eLogika. Jeho možnosti, využití a základní entity.

Další část popisuje obecně data mining, historii vývoje, definice. Následuje vysvětlení metodologií CRISP-DM a SEMMA pro tvorbu data miningových řešení. Kapitola se dále zabývá využitím data miningových metod v marketingu a e-learningu.

Třetí část diplomové práce se zabývá popisem data miningových metod obsažených v SQL Serveru 2008. Dále se v kapitole věnujeme jazyku DMX, pomocí kterého se vytvářejí dolovací struktury a dolovací modely.

V závěrečné kapitole je podrobně popsán vývoj a nasazení vybraných data miningových metod do systému eLogika. Tato kapitola je strukturována pomocí jednotlivých fází metodologie CRISP-DM, která byla vybrána jako vhodná pro tvorbu data miningového procesu.

2 E-learningový systém eLogika

Systém eLogika je řídící výukový systém v originální terminologii Learning Management System (LMS), který spravuje administrativu a organizaci výuky v rámci e-learningu (1). Základními funkcemi LMS systémů jsou například vytváření a správa kurzů, evidence studentů, výukových materiálů atd. Součástí LMS systémů jsou také funkce pro ověřování studentem nabytých znalostí pomocí vytváření testů a následného přezkoušení.

Systém eLogika je webová aplikace která využívá e-learningovou formu vzdělávání. Jedná se o využití moderních informačních a komunikačních technologii v procesu výuky (2). Tato forma vzdělávání sebou přináší řadu výhod jako je například nezávislost studenta na místě a čase výuky, jednoduchá dostupnost studijních materiálů, možnost průběžného ověřování znalostí atd. Mezi nevýhody naopak patří absence osobního kontaktu mezi vyučujícím a studentem, která sebou přináší velmi dlouhou prodlevu při řešení studentových problémů s látkou, nutnost dostupnosti k vybavení pro používání e-learningu, atd.

Základním účelem systému eLogika je zjednodušení práce vyučujícím z důvodu zvyšujícího se počtu studentů. Vyučující se nemohou osobně věnovat studentům v dostatečně možné míře, a proto tuto funkci nahrazuje tento e-learningový systém. Jak již bylo řečeno, jedná se o webovou aplikaci, která sebou přináší řadu funkcí spojených se správou vyučovaných kurzů, evidenci studentů, vytváření testů atd.

Vzhledem k zaměření této diplomové práce je vhodné si obecně popsat základní entity obsažené v systému eLogika z důvodu možností zmiňování v textu:

Škola – Základní entita systému, popisuje evidovanou školu, která následně obsahuje další administrativní entity jako je akademický rok, semestr, kurz, třídy atd.

Akademický rok – Základní časová jednotka, která popisuje délku trvání jednotlivých akademických intervalů, například září 2010 – srpen 2011.

Semestr – Podrobnější časové rozdělení akademického roku na základě administrativní struktury školy, nejčastěji se vyžívá rozdělení zimní a letní semestr.

Kurz – Jedná se o e-learningové kurzy vyučované v systému eLogika. Kurz je vytvářen obecně. Každá další jeho instance je zaváděna v rámci semestru.

Garant, tutor – Každý kurz má svého garanta a ostatní tutory. Pro jednoduchost se tyto dvě role systému v textu jednotně označují jako vyučující.

Student – Role systému eLogika, jak napovídá název, jedná se o studenta určitého kurzu.

Třída – Rozdělení studentů určitých kurzů do jednotlivých tříd. Každou třídu spravuje jeden tutor.

Podmínky kurzu – Pro jednotlivé kurzy lze zavést různé podmínky splnění tohoto kurzu. Jedná se o tzv. skupiny aktivit, které obsahují cvičné nebo hlavní testy. Ty musí student splnit, aby vyhověl podmínkám kurzu.

Kapitola – Tato entita popisuje základní rozdělení vyučované látky v rámci kurzu.

Kategorie – Rozdělní kapitol na dílčí části. Kategorie obsahují otázky, ze kterých budou následně generovány testy pro ověření znalostí studentů.

Otázka – Jak již bylo řečeno, jedná se o otázku na určitou látku kategorie. Systém eLogika eviduje dva druhy otázek a to formulářové, na které studen odpovídá slovně, a dále otázky se správnými, resp. špatnými odpověďmi, které student následně vybírá dle svého uvážení. Dalším rozdělení otázek je na hlavní, cvičné a ukázkové.

Šablona testu – Jedná se o tzv. strukturu testu, na základě které bude vygenerován konkrétní test. Tuto šablonu zavádí vyučující. Skládá se z bloků, ve kterých se nastavují různé kriteria pro počet otázek, typ hodnocení, počet odpovědí k jednotlivým otázkám atd. Tuto šablonu lze využít pro generování více testů.

Test – Na základě šablony testu lze vytvořit konkrétní test. Ty se stejně jako otázky rozdělují na hlavní, cvičné a ukázkové. V případě cvičných testů jsou při generování testu vybírány pouze cvičné otázky. Stejně tak u hlavního testu se generuje test pouze z hlavních otázek. Vyučující nastaví parametry jako minimální / maximální počet bodů, počet pokusů, čas vypracování aj. Systém eLogika eviduje dva druhy testů, první je papírový, které lze po vygenerování vytisknout, a studenti ho mohou vypracovat v učebně. Druhý je tzv. on-line test, který studenti vypracovávají přes webové rozhraní systému eLogika.

Ostatní funkce a entity systému eLogika jsou podobně popsány v diplomových pracích těchto studentů:

Bc. Tomáš Loskot – „Analýza pro systém eLogika“

Bc. Vojtěch Hernas – „Systém eLogika a e-learningová podpora výuky“

3 Data mining

V kapitole jsem vycházel z této literatury: (3), (4), (5)

Pojem data mining, neboli jeho obecný český překlad dolování dat, se nakonec staly základním popisem nového oboru informatiky. O původním odvětví dobývání znalostí z databází (KDD - Knowledge Discovery in Databases) se začalo mluvit v akademických kruzích již počátkem devadesátých let. Hlavním impulzem vývoje bylo množství databázově uchovávaných dat. Původní manuální techniky výzkumu se s přibývajícím množstvím těchto dat stávaly nereálné a tak daly možnost vzniku tomuto novému oboru. Současně s tím vznikl i zájem velkých Amerických firem, které chtěly svá obsáhlá data využít ve strategickém rozhodování. Velkou pozornost tomuto odvětví dokládají i vzniklé odborné konference (např. DMIN - International Conference on Data Mining - Mezinárodní konference o data miningu, ECML-PKDD - European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases - Evropská konference strojového učení a zásady a postupy pro získávání znalostí z databází), odborné skupiny (SIGKDD - Special Interest Group on Knowledge Discovery and Data Mining), nebo odborné časopisy (časopis Data mining and Knowledge Discovery). Tyto předpoklady napomohly v nárůstu zájmu odborné komunity o obor dobývání znalostí z databází.

Citace:

„KDD is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data” (Fayyad, Piatetsky-Shapiro, and Smyth 1996).

Překlad:

Dobývání znalostí z databází (KDD) lze definovat jako netriviální extrakci implicitních,
dříve neznámých a potenciálně užitečných informací z dat.

Oproti prostému použití statistických metod a metod strojového učení se v procesu dobývání dat více klade důraz na předzpracování dat k analýze a na interpretaci výsledných znalostí.

V současné době mnoho organizací řeší své vnitro-firemní procesy pomocí různých informačních systémů a to sebou přináší i využívání databází a datových skladů. Během několika let provozu si vytvořily rozsáhlé databáze, které již neodrážejí jen současný stav, ale lze v nich dohledat i historii. Taková data mohou skrývat souvislosti a vzory chování, jejíž znalost může být potencionálně užitečná.

V následující podkapitole bude popsáno využití metodologií pro aplikaci data miningových metod na určitý problém a co to metodologie vlastně je. Následně popíšu dvě nejznámější metodologie.

3.1 Metodologie Data miningu

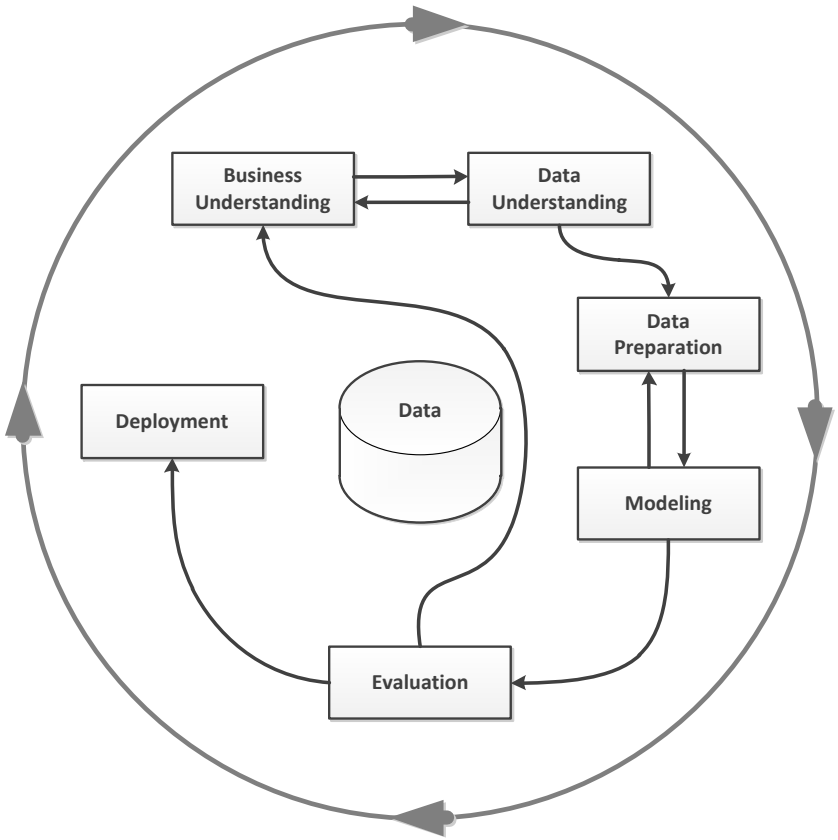
Cílem metodologie je popsat univerzální postup pro řešení určitých úloh z oblasti data miningu, který bude použitelný v nejrůznějších případech. Tento postup pomáhá vyhnout se běžným chybám při tvorbě data miningového projektu.

Data mining je získávání hodnotných informací ve velkých objemech dat, která vyžaduje velké

množství lidských, hmotných, datových a softwarových zdrojů, které jsou velmi nákladné. Za účelem snížení nákladů vznikala snaha tyto metody provádět standardizovaným postupem. Proto byly vytvořeny metodologie, které popisují efektivní postup při řešení projektů. Mezi základní metodologie patří například CRISP-DM, SEMMA, Java Data Mining (JDM).

3.1.1 CRISP-DM

Vývoj metodologie CRISP-DM (CRoss Industry Standard Process for Data Mining) byl zahájen jako projekt Evropské komise definující model standardního postupu při vytváření data miningových projektů. Metodologie CRISP-DM byla vytvořena konsorciem firem NCR systems Engineering Kopenhagen, SPSS Inc., DaimlerChrysler a OHRA Verzekering en Bankk Groep B.V. Jedná se o metodologii nezávislou na volbě software.



Obrázek 3.1 - Životní cyklus projektu CRISP-DM

Schéma na obrázku popisuje životní cyklus metodologie CRISP-DM. Ačkoli jsou zde vyznačeny určité ideální toky, pořadí jednotlivých fází není striktně dáno. Výsledek jedné fáze ovlivňuje volbu kroků dalších, proto je potřeba se k určitým fázím neustále vracet. Tento cyklický postup vyznačuje vnější kruh. Středem schématu jsou analyzovaná data, která jsou základem data miningového projektu a kolem kterých se tento proces neustále točí.

Metodologie CRISP-DM rozděluje data miningový proces do šesti základních fází:

- **Business Understanding** – Porozumění problematice. Základem projektu je pevné stanovení cílů a požadavků. Je třeba převést tyto cíle do problému dolování dat a

následně vypracovat předběžný plán zaměřený na dosažení těchto cílů. Abychom správně pochopili data, která budou následně analyzována, je nezbytné porozumět firemním procesům, pro které je řešení hledáno.

- **Data Understanding** – Porozumění datům. Tato fáze se skládá ze čtyř základních kroků. Začíná počátečním sběrem analyzovaných dat, následuje popis dat, průzkum dat a ověřování kvality dat. Po provedení těchto kroků získá datový analytik základní představu o datech a bude schopen posoudit kvalitu těchto vstupních dat, popsat data, popř. odhalit zajímavé podmnožiny dat pro následné testování hypotéz.
- **Data Preparation** – Příprava dat. Cílem této fáze je vytvoření datového souboru (datového skladu) pro následnou analýzu. Tomuto kroku je třeba věnovat velkou pozornost, aby data byla dostatečně reprezentativní a očištěná od případných nežádoucích chyb. Jako nejvhodnější pro předzpracování dat se jeví využití procesů ETL.
- **Modeling** – Modelování. V této fázi jsou na předpřipravená data aplikovány stávající data miningové metody. Obvykle existuje několik technik pro řešení určité úlohy, proto je třeba vybrat tu nejvhodnější. Doporučuje se využít více metod a jejich výsledky následně kombinovat. Některé techniky mají specifické požadavky na vstupní formy údajů, proto se neustále vracíme k předcházející fázi Příprava dat. Jedná se o iterační proces, kdy aplikujeme určité algoritmy s různými vstupními parametry.
- **Evaluation** – Vyhodnocení výsledků. Před konečným nasazením modelu je potřeba důkladně vyhodnotit, zda bylo dosaženo požadovaných cílů. Na výsledcích se ověřuje, zda se neopomněl nějaký důležitý faktor, který by mohl ovlivnit výsledky. Pokud je model vyhodnocen jako dostačující, přechází se na Implementaci.
- **Deployment** – Implementace (Zprovoznění). Získané znalosti je třeba interpretovat, aby se daly konkrétně využívat. Z tohoto důvodu se nejčastěji zavádí tzv. živé modely pro opětovné použití a získání nových informací a znalostí.

3.1.2 SEMMA

Další možnou variantou je metodologie SEMMA, za kterou stojí společnost SAS. Přesněji se nejedná přímo o metodiku data mining, ale spíše o logické uspořádání funkčních nástrojů SAS Enterprise Miner, pro provádění klíčových úkonů.

Zkratka SEMMA je složena z těchto pěti kroků:

- **Sample** - Vybírání vhodných objektů. Pokud je vstupem rozsáhlý datový soubor, přistupuje se k výběru statisticky reprezentativního vzorku dat. Analyzování tohoto vzorku snižuje dobu zpracování potřebnou k získání informací a znalostí.
- **Explore** - Vizuální explorace a redukce dat. V tomto kroku se zkoumá výběrový vzorek dat. Hledají se neočekávané hodnoty, anomálie, zjišťují se vlastnosti a charakteristiky. Využívá se klasických statistických postupů (minima, maxima, průměr, medián, modus, směrodatná odchylka, rozptyl, odlehlá pozorování). Průzkum dat pomáhá zdokonalit proces objevování.
- **Modify** - Seskupování objektů a hodnot atributů, datové transformace. Kontrola správnosti atributů a případná modifikace vstupního souboru. Hledání podskupin,

různých seznamů a zavádění nových proměnných. Výstupem by měl být soubor bez odchylek a extrémů.

- **Model** - Analýza dat. Tvorba data miningového modelu pro získání požadovaného výsledku. Aplikování různých metod pro získání znalostí a informací.
- **Assess** - Porovnání modelů a jejich interpretace. Vyhodnocení získaných výsledků podle užitečnosti a spolehlivosti.

3.2 Oblasti využití data miningu

Různé metody data miningu se využívají k jednotnému cíli a to najít v datech závislosti a vzory chování. Tyto závislosti lze označit jako prediktivní a předpokládat tak, že pokud se určitý vzor chování opakoval již v minulosti, bude se takto chovat i v budoucnu. Vzory v určitém chování se vyhledávají už od nepaměti. Mnohé z nich používáme denně, aniž si to uvědomujeme, a to například v podobě pranostik. Jedná se o lidovou průpovídku, která popisuje předpověď počasí nebo životní zkušenost. Mezi jednu z nejznámější patří např. „Březen - za kamna vlezem; duben - ještě tam budem.“. Lidé si ze vzpomínek z historie počasí našli určitý vzor a touto rýmovanou pranostikou si ulehčují jejich zapamatování. Bohužel tento vzor nelze pokaždé brát jako jistotu a v dubnu počítat s tím, že se stále budeme ohřívat u oněch kamen. Obdobný případ nastává i ve vzorech nalezených pomocí data miningu. Ty mohou sloužit jako určitá výhoda pro přesnější rozhodnutí, ale nelze je brát jako zaručenou věc.

Data mining je pouze prostředek, pomoci kterého lze získat informace pro podporu rozhodování. Samotnou úvahu a rozhodnutí musí už udělat příslušná zodpovědná osoba. (6)

Díky data miningu lze studovat, porozumět a podle všeho i vylepšit v podstatě jakýkoli proces v navzájem velmi odlišných oblastech. Mezi tyto oblasti patří například pojišťovnictví, bankovnictví, telekomunikace, maloobchod, cestovní ruch, lékařství.

Nejčastěji se data mining využívá v marketingovém odvětví, konkrétně v oblasti styku se zákazníkem (CRM), ale také i v oblasti kontroly kvality produktů (6). Právě v marketingu zažil data mining největšího rozmachu. Motivace byla zřejmá, a to zvýšení prodejnosti produktů a s tím spojené navýšení zisků. Právě lépe cílené reklamní kampaně mohou mít za důsledek přilákání nových zákazníků a snížení nákladů. Nejenom nově příchozí zákazník vytváří profit, ale z dlouhodobého pohledu je mnohem podstatnější udržení stávajících zákazníků. Využitím modelů pro identifikaci zákazníků s vyšší pravděpodobností odchodu ke konkurenci lze tomuto jevu předcházet a zavčas podniknout kroky pro jeho udržení. Zpravidla je mnohem levnější si zákazníky udržovat, než je znovu obtížně získávat zpět. Jedním ze způsobu zabránění odchodu zákazníka je zvýšení jeho spokojenosti. V tvrdém konkurenčním prostředí je spokojenost zákazníka základním kamenem úspěchu.

Dalším oborem využití je například maloobchod pro zkoumání faktorů, které významně ovlivňují nákupní chování, zjišťování asociací nakupovaného zboží a následná optimalizace rozložení zboží v obchodě. V bankovnictví se data mining využívá při rozhodování o poskytnutí půjčky vzhledem k historii a schopnosti splácet, dále pro predikci nepovolených transakcí s odcizenými platebními kartami. Ve vědeckém výzkumu například při analýze genetických informací, nebo v zabezpečení při monitorování aktivit v systému s cílem odhalit potencionální škůdce.

3.3 Rozdílnost použití v marketingu a e-learningu

Již na začátku projektu je třeba si stanovit základní cíle, kterých chceme dosáhnout. Zatím, co v marketingu je hlavní motivací navýšení prodejnosti produktů resp. zisků, u e-learningu je hlavním stimulem zvýšení úspěšnosti studentů v jejich cestě studiem. Pro lepší srovnání rozdílnosti přístupu v marketingu a e-learningu lze zavést pojem e-commerce. Oblasti e-learningu se budeme podrobněji zabývat v dalších kapitolách. Pojem e-commerce popisuje obchodování na internetu a zahrnuje všechny jeho podstatně části (e-shop včetně vlastních databází, elektronické online platby, SEO), které následně slouží jako zdroje dat pro data mining. Výsledky z těchto metod lze použít pro lepší marketingová rozhodnutí v tomto oboru.

Důležité rozdíly a přístupy při použití data miningových metod v e-learningu a e-commerce:

- **Okruh** – e-commerce má účel provést klienta v nákupu, zato e-learning provést studenta látkou.

- **Data** – v e-commerci se používají standardní data o prodeji zboží, počtu kusů, obsahu nákupního košíku atd. V e-learningu je více informací o studentově práci se systémem, jeho výsledky v testech, popř. studentova interakce.

- **Účel** – využití data miningu v e-commerce je založeno na zvýšení profitu, který je hmatatelný a může být měřitelný ziskem. Data mining využívaný v e-learningových systémech má za cíl zlepšování výuky a s tím vyšší úspěšnost studentů.

- **Technika** – Výukové systémy mají speciální charakteristiku a vyžadují jiné postupy dolování znalostí. Důsledkem je, že některé specifické data miningový metody budou použity na jednotlivých procesech výuky.

Ačkoli je využití dolování dat v marketingu a e-learningu v některých částech obdobné, najdou se zde i rozdíly.

3.4 Software

V současné době existuje široká nabídka specializovaných softwarů určených pro data mining. Mezi nekomerční aplikace patří Weka, RapidMiner nebo Orange. Nejznámější komerční aplikace jsou například SAS Enterprise Miner, IBM SPSS Modeler, STATISTICA Data Miner nebo SQL Server 2008.

3.4.1 Weka

Weka (Waikato Environment for Knowledge Analysis) je populární software vyvíjený na Univerzitě Waikato na Novém Zélandu. Weka je software vyvíjený v jazyce Java pod licencí GNU General Public License. Balík obsahuje nástroje pro předzpracování dat, klasifikaci, regresi, shlukování, asociační pravidla a vizualizace. Softwaru Weka využívá kupříkladu známý komplexní balík business softwaru Pentaho, který obsahuje různé spolupracující aplikace řešící jednotlivé segmenty business intelligence.

3.4.2 RapidMiner

Software RapidMiner, který byl dříve známý pod názvem YALE (Yet Another Learning Environment), se prioritně zabývá oblastí marketingu a finančnictví. RapidMiner je vyvíjen v jazyku Java, proto funguje na všech hlavních platformách a operačních systémech. Jedná se o open source systém. Obsahuje základní moduly pro integraci dat, ETL procesy, analýzu dat, reportovací služby. Pomocí speciálního rozhraní lze v RapidMineru zpřístupnit i algoritmy ze systému Weka.

3.4.3 Orange

Produkt Orange je komponentní knihovna jazyka Phyton vyvinuta v Laboratořích umělé inteligence, Fakulty informatiky a informačních věd Univerzity v Lublani ve Slovinsku. Orange je bezplatný software, který lze používat a upravovat podle podmínek GNU General Public License. Velmi jednoduché a přehledné uživatelské rozhraní Orange tvoří tzv. uzly (widget), které se neustále vyvíjejí a doplňují.

3.4.4 SAS Enterprise Miner

Software SAS Enterprise Miner, který je produktem firmy SAS Institute, umožňuje vytvářet procesy dolování dat a tvorbu prediktivních a deskriptivních modelů založených na analýze rozsáhlých dat. Obsahuje i modul, který lze vložit do tabulkového procesoru Microsoft Excel pro jednodušší používání.

3.4.5 IBM SPSS Modeler

Společnost IBM se nedávno rozrostla o analytický software firmy SPSS. Výsledkem této akvizice je produkt IBM SPSS Statistics, jehož součástí je nástroj pro data mining IBM SPSS Modeler. Software obsahuje velké množství předpřipravených algoritmů pro vytváření modelů. Modely lze interaktivně zobrazovat a zároveň používat různé vizualizační techniky pro prezentování výsledků.

3.4.6 STATISTICA Data Miner

Jedná se o českého zástupce software pro data mining. STATISTICA Data Miner je součástí systému STATISTICA. Zahrnuje velký výběr algoritmů pro shlukování, interaktivních regresních a klasifikačních stromů, různé typy neuronových sítí. Software STATISTICA Data Miner obsahuje již předpřipravené projekty, které lze aplikovat na vlastní data miningový problém.

3.4.7 SQL Server 2008

Databázový server společnosti Microsoft je v rámci platformy SQL Server Business Intelligence (BI) rozdělen do několika základních bloků. Jedním z těchto bloků je SQL Server Analysis Services, který obsahuje data miningové metody a další analytické služby. Podrobněji se SQL Serveru 2008 a jeho částem budeme věnovat v dalších kapitolách.

4 SQL Server 2008 a data mining

V kapitole jsem vycházel z této literatury: (3), (5), (6), (7), (8), (9), (10)

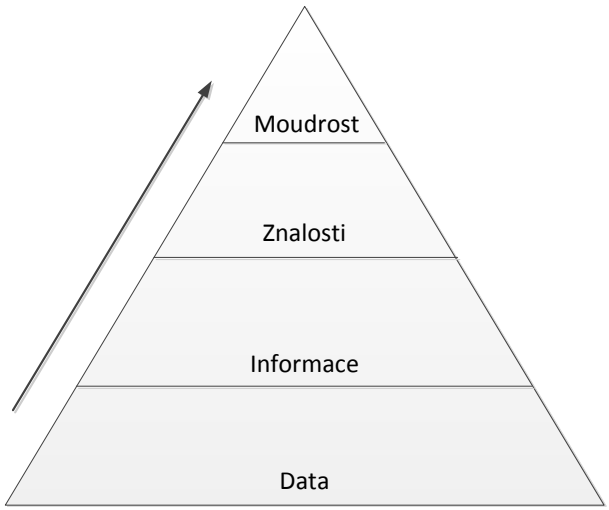
Microsoft SQL Server 2008 je již třetí verze SQL Serveru, který je dodáván s technologiemi pro dolování dat. Postupem času se data mining v SQL Serveru rozrostl z původně dvou algoritmů obsažených v SQL Server Analysis services (Analytické služby) na nedílnou část SQL Server Business Intelligence, která je plně integrovaná s OLAP, Integration Services (Integrační služby), a Reporting Services (Reportovací služby) (7).

4.1 Business Intelligence

V roce 1989 Howard Dresner ze společnosti Gartner Group definoval termín Business Intelligence (BI) takto:

Business Intelligence je množina konceptů a metodik, které zlepšují rozhodovací proces za použití metrik, nebo systémů založených na metrikách. Účelem procesu je konvertovat velké objemy dat na poznatky, které jsou potřebné pro koncové uživatele. Tyto poznatky potom můžeme efektivně použít například v procesu rozhodování a mohou tvořit velmi významnou konkurenční výhodu.

V dnešním vysoce konkurenčním prostředí, kdy je neustále potřeba dělat správná rozhodnutí, je Business Intelligence nástroj, který může s touto nelehkou úlohou pomoci. Základem kvalitního rozhodnutí jsou kvalitní informace, které má manažer k dispozici ve správný čas, na správném místě a v potřebné podobě (6).



Obrázek 4.1 - Informační hierarchie

Základním kamenem jsou data. Ta obsahují pouze jednoduchá fakta, u kterých se předpokládá, že jsou v nich ukryty informace. Pokud se k těmto datům přidají i souvislosti, mohou nám informace vyplout na povrch. Třetím stupněm v informační hierarchii jsou znalosti, které vznikají ze širší aplikace vzdělání, zkušeností a odbornosti na vyhodnocení informací. Následné zhodnocení znalostí a jejich uplatnění přináší moudrost. Moudrost lze popsat jako pochopení jevů a událostí, zákonitostí a principů v jejich celé hloubce i šířce z pohledu minulosti,

současnosti ale i budoucnosti.

Jinými slovy Business Intelligence je soubor procesů, aplikací a technologií, jejímž cílem je účelně a účinně podporovat rozhodování ve firmě. Mezi tyto nástroje patří například datové sklady (data warehouse), OLAP (On-line Analytical Processing), ale také data mining.

4.2 Jazyk DMX

Data Mining Extensions (DMX) je dotazovací jazyk SQL Serveru 2008, který je primárně určen pro práci s funkcemi data miningu. DMX byl poprvé představen v OLE DB pro data mining specifikaci v roce 1999. Cílem této specifikace bylo vytvořit neutrální programovatelné rozhraní, které využívá již známé koncepty. Vývojáři se snažili vytvořit dotazovací jazyk blízký SQL, ale aby zároveň uspokojoval potřeby dolování dat. Postupem času se cílem stalo zahrnutí .NET vývojářů, kteří používají C# nebo VB.NET, takže význam OLE DB klesal. Stále ale zůstává flexibilní a popisný jazyk DMX, který umožňuje jednoduchou implementaci data miningových řešení (7).

Data miningový scénář je jako pečlivě formulovaná otázka s velmi specifickými pravidly, jak se na ni zeptat. Proto je potřeba popsat si klíčové pojmy, abychom dokázali úspěšně formulovat otázku a provést data miningový scénář.

4.2.1 Atribut (Attribute)

Nejzákladnější jednotka dolování dat je atribut, jedná se o informaci, charakteristiku, vlastnost entity. Entitou rozumíme jakoukoli osobu, zvíře, věc nebo objekt reálného světa. Například entita zákazník zahrnuje atributy pohlaví, věk, rodinný stav, nebo entita produkt obsahuje atributy kategorie, množství, barva atd. Při výběru atributů pro dolování dat je potřeba pečlivě volit, které jsou relevantní pro případné otázky a poskytnou validní informace, které mohou být použity v data miningové metodě.

Mnohé atributy nejsou vhodné pro dolování dat, nebo alespoň ne v podobě, ve které jsou uloženy. Například u atributu ulice, který je součástí adresy entity zákazníka, nelze vytvářet korelace k dalším atributům, dokud nebude transformován pomocí geoprostorových nástrojů pro určení vzdálenosti ke konkrétnímu orientačnímu bodu (např. vzdálenost od města). Obecně se doporučuje vytvořit co nejvíce specifické atributy pro následné lepší položení otázky. Další možností získání informací z dat je odvození atributů. Kupříkladu hodnota maximální částky, kterou zákazník utratil nebo počet týdnů, kdy byl výrobek k prodeji. Je potřeba kreativně pohlížet na aspekty vlastního problému a přeměnit dosavadní data na bohatství. S tím jak roste počet atributů, zvyšuje se i množství kombinací, ze kterých lze získat další užitečné vzory, a přesně toto je cílem data miningu.

4.2.2 Případ (Case)

Případ patří k nejdůležitějším pojmům DMX. Jedná se o jeden příklad určitého sledování, který se poskytuje algoritmu pro dolování dat. I když to zní jednoduše, špatné pochopení sebou přináší nejčastější selhání v aplikaci data miningových metod. Případ představuje entitu, která se „doluje“, a na kterou se vytváří otázka. V nejjednodušším příkladě (což bývá ve skutečnosti většina případů) reprezentuje případ jeden řádek tabulky a sloupce reprezentují jednotlivé

atributy.

I v jednoduchých situacích může vzniknout nepochopení, jak by měl být případ reprezentován. Například kdybychom chtěli porozumět, jaké faktory ovlivňují kriminalitu ve městech. Nabízí se, že v tomto příkladě bude případem město, resp. atributy o něm, dopady zločinu. Nicméně dalším zvážením situace, chceme zjistit, jak město samotné přispívá k celkové kriminalitě. Z toho důvodu je město také atributem, a proto se stává jednou z nezávislých proměnných, které budou použity k analýze. V této situaci bude případem něco jako měření, které bude obsahovat název města, kriminalitu a další atributy.

Jeden ze způsobů, jak správně identifikovat případy je přemýšlet, co je v tomto případě anonymní. Pokud budeme analyzovat faktory zákazníků, pak je zákazník pravděpodobně naším případem, protože nezáleží na konkrétních zákaznících, ale pouze na jejich vlastnostech v porovnání s vlastnostmi jiných zákazníků.

Složitější situace nastává, když chceme analyzovat vztahy mezi atributy, které mohou nebo nemusí existovat v rámci případu a počet atributů není znám předem. Společným příkladem je tzv. „Analýza nákupního košíku“, kdy chceme zjistit, které výrobky se v rámci jednoho nákupu prodávají společně a které produkty ve výprodeji předpovídají prodej jiných produktů. V tabulce máme sloupec představující ID transakce, sloupec Produkt, Množství a sloupec Výprodej.

ID transakce	Produkt	Množství	Výprodej
1	Rohlík	6	ne
1	Mléko	2	ne
2	Chleba	1	ne
2	Pivo	2	ano
3	Mléko	4	ne
3	Chleba	1	ne
3	Televize	1	ano

Tabulka 4.1 - Tabulka transakcí

V tabulce transakce, všechny řádky patřící ke stejné transakci (identifikované podle ID transakce) obsahují případy. Každý z těchto případů pak má vlastní atributy jako jsou Produkt, Množství a Výprodej. Pokud se takováto transakční tabulka objeví, nazývá se vnořená tabulka (nested table). Stejně tak, pokud nějaký případ obsahuje vnořenou tabulku, označuje se vnořeným případem (nested case).

4.2.3 Klíče (Keys)

Jazyk DMX zahrnuje dva různé typy sloupců jako klíče. Tyto sloupce mají dva odlišné významy a důsledky. První je klíč případu (case key), který označuje identifikaci entity zastoupené v případu. V mnoha implementacích lze primární klíč zdrojových dat použít jako klíč případu. Často pokud data neobsahují vnořené tabulky, lze jako klíč případu použít index řádku nebo jiný zvolený identifikátor.

Druhým typem klíče je tzv. vnořený klíč (nested key), který je zcela odlišný. Zatímco klíč případu představuje anonymní část případu, vnořený klíč označuje název subjektu vnořeného řádku.

				Nákupy	
ID	Jméno	Pohlaví	Věk	Produkt	Množství
1	Jan	Muž	35	Mléko	2
				Chleba	1
				Rohlík	4

Tabulka 4.2 – Vnořený případ zákazník

V ukázkové tabulce je klíčem vnořené tabulky Nákupy atribut Produkt, což znamená, že atribut Množství se odkazuje na položku Produkt v tomto sloupci.

Častou chybou je použití cizího klíče jako vnořeného klíče. Stává se to nejčastěji z důvodu pojmenování. Uživatele svádí označit tento klíč jako vnořený klíč z pohledu relační databáze. Nicméně vnořený klíč je pouze klíč v rámci vnořené tabulky pro jednu entitu.

4.2.4 Vstupy a Výstupy (Inputs and Outputs)

Každý atribut v DMX jazyku může být vstupní, výstupní nebo obojí. Tato koncepce se zdá být poněkud jednoduchá, ale v praxi to může přinášet řadu nejasností. V obecné rovině data miningové algoritmy používají vstupní atributy, aby se dozvěděly něco o těch výstupních. Nicméně každý algoritmus odvozuje svou vlastní interpretaci co se přesně vstupem a výstupem rozumí. Tomuto problému se budeme věnovat v další části této kapitoly, kde si popíšeme jednotlivé data miningové algoritmy.

V případě, že je atribut označen zároveň jako vstupní a výstupní, podle přijaté konvence Microsoft algoritmů se nestane, že by vstupní atribut předpovídal sebe sama. Algoritmy pak automaticky podniknou kroky k řádnému odseparování informací. Možnost mít více výstupů v jednom modelu je inovace v DMX, která sebou přináší možnost vytváření mnohem komplexnějších a zajímavějších scénářů.

4.2.5 Dolovací struktura (Mining structure)

Dolovací struktura (nebo jinými slovy miningová struktura) popisuje podobu problému. Představuje sloupce dat, které jsou k dostupné pro data mining problém, popis těchto dat a zdroje dat pro testování daných dolovacích modelů. Pojem dolovací model bude vysvětlen v další části. Každý z těchto sloupců musí obsahovat popis datového typu, a zda se jedná o kategorickou nebo spojitou proměnou, což popisuje, jakým způsobem se bude s touto proměnou manipulovat.

Je běžnou praxí, že dolovací struktura obsahuje více sloupců, než budou následně použity v jednotlivých modelech. Lze například v rámci jedné struktury vytvořit mnoho různých modelů s použitím odlišných data miningových algoritmů, pro řešení konkrétního problému dolování dat. Vzhledem k tomu, že algoritmy mají různé funkce, mají i různé požadavky na vstupní data. Například algoritmus lineární regrese akceptuje pouze spojité data. Oproti tomu naivní Bayesova metoda přijímá pouze kategorická data. V situacích, kdy by bylo potřeba využít obou těchto algoritmů na stejném zdroji dat, sloupce jednoho by se neslučovaly s druhým. Z tohoto důvodu lze vytvořit kategorickou i spojitou verzi sloupce, které budou vázány na stejný zdroj dat a následně použity v příslušném algoritmu.

Vhledem k tomu, že jazyk DMX byt navržen tak, aby se co nejvíce podobal jazyku SQL, je i

tvorba dolovací struktury velmi podobná vytváření tabulek.

```
CREATE MINING STRUCTURE [Zakaznik1]
(
      [IDZak]     LONG  KEY,
      [Jmeno]     TEXT  DISCRETE,
      [Pohlavi]   TEXT  DISCRETE,
      [Vek]       LONG  CONTINUOUS,
      [Auto]      TEXT  DISCRETE,
      [AutoTyp]   TEXT  DISCRETE

)
```

Ze syntaxe lze snadno identifikovat, že byla vytvořena dolovací struktura s názvem `Zakaznik1`, která obsahuje sloupce `IDZak`, `Jmeno`, `Pohlavi`, `Vek`, `Auto`, `AutoModel`. Sloupec `IDZak` je klíčem `KEY` dolovací struktury a je stejně jako sloupec `Vek` označen jako číslo. Ostatní sloupce jsou označeny jako text. Druhým atributem je typ obsahu sloupce, v našem případě kategorické `DISCRETE` (diskrétní) anebo spojité `CONTINUOUS`.

Další možností je označit sloupec jako `DISCRETIZED`, což lze volně přeložit jako diskretizovaný. Jedním z důvodu použití tohoto nastavení je, že určité rozmezí hodnot dává ve výsledku větší smysl při sledování vzorů. Dalším důvodem je, že některé algoritmy nepodporují spojité hodnoty. Typicky se nastavení rozmezí a kategorizace provádí při předzpracování dat. Nicméně může nastat i situace, kdy nelze dopředu předvídat, v jakém rozmezí se budou hodnoty pohybovat. Z tohoto důvodu lze využít SQL Server pro automatickou diskretizaci, který pak sám zváží rozložení hodnot a určí ideální kategorie.

Výchozí nastavení systému funguje způsobem, že se hodnoty rovnoměrně rozdělí do pěti kategorií. Pokud tato volba selže, přejde systém ke kategorizaci pomocí shluků. Možné parametry nastavení `DISCRETIZED` (diskterizace):

- `EQUAL_AREAS` – Systém vypočítá rozmezí tak, aby v každé části (dle nastaveného počtu částí) byl stejný počet hodnot.
- `CLUSTERS` – Použití jednorozměrného shlukování k nalezení seskupených oblastí.
- `AUTOMATIC` (Výchozí) – Systém nejprve použije `EQUAL_AREAS` a pokud tato volba neuspěje, přejde k použití `CLUSTERS`.

```
CREATE MINING STRUCTURE [Zakaznik2]
(
      [IDZak]     LONG  KEY,
      [Jmeno]     TEXT  DISCRETE,
      [Pohlavi]   TEXT  DISCRETE,
      [Vek]       LONG  CONTINUOUS,
      [VekDisc]   LONG  DISCRETIZED(EQUAL_AREAS,3),
      [Auto]      TEXT  DISCRETE,
      [AutoTyp]   TEXT  DISCRETE

)
```

V ukázce lze vidět diskretizovaný sloupec `VekDisc`.

Vytvoření dolovací struktury obsahující vnořenou tabulku se provádí pomocí datového typu `TABLE`. Vnořená tabulka je definována obdobným způsobem jako samotná struktura.

```
CREATE MINING STRUCTURE [Zakaznik3]
(
      [IDZak]     LONG  KEY,
      [Jmeno]     TEXT  DISCRETE,
      [Pohlavi]   TEXT  DISCRETE,
      [Vek]       LONG  CONTINUOUS,
      [VekDisc]   LONG  DISCRETIZED(EQUAL_AREAS,3),
      [Auto]      TEXT  DISCRETE,
      [AutoTyp]   TEXT  DISCRETE,
      [Nakupy]    TABLE
      (
            [Produkt]   TEXT        KEY,
            [Mnozstvi]  LONG        CONTINUOUS,
            [Vyprodej]  BOOLEAN     DISCRETE
      )
)
```

Ukázka dolovací struktury s vnořenou tabulkou.

4.2.6 Dolovací model (Mining model)

Vzhledem k tomu, že dolovací struktura je něco ve smyslu obalu, dolovací model je její objekt, který převádí řádky dat na případy a provádí strojové učení pomocí určitého data miningového algoritmu. Dolovací model lze popsat jako podmnožina sloupců v dolovací struktuře, které budou použity jako atributy (včetně nastavení vstupní, výstupní nebo obojí) spolu s algoritmem dolování dat a jeho parametry. Struktura dat se může libovolně filtrovat a dolovat pouze data vhodná pro konkrétní problém. Například pokud chceme vymezit jen pro určitý region, výrobky konkrétní kategorie nebo jinou oblast zájmu. Po zpracování dolovacího modelu, bude tento model obsahovat vzory, které data miningový algoritmus odvodil z dat.

Pro vytvoření dolovacího modelu pomocí jazyku DMX je nutné specifikovat podmnožinu sloupců a klíč dolovací struktury popř. klíč vnořené tabulky. Nejjednodušším způsobem je vložit do dolovací struktury výchozí model společně s názvem data miningového algoritmu.

```
ALTER MINING STRUCTURE [Zakaznik1]
ADD MINING MODEL [ZakaznikClusters]
USING Microsoft_Clustering
```

V tomto případě budou všechny sloupce považovány za vstup, proto je tento postup vhodný pouze pro metodu shlukování. Tento algoritmus totiž nevyžaduje výstup, protože výstupem jsou názvy shluků.

Ve většině případů je však seznam sloupců vyžadován, protože je nutné zadat výstupní sloupce.

Nastavení sloupce jako vstupní nebo výstupní se provádí pomoci parametru `PREDICT` a `PREDICT_ONLY`. U sloupce, který je nastaven bez parametru se jedná a sloupec vstupní. Pokud je sloupec nastaven jako `PREDICT`, jedná se o sloupec vstupní i výstupní. Sloupec nastavený parametrem `PREDICT_ONLY` je pouze výstupním sloupcem.

```
ALTER MINING STRUCTURE [Zakaznik2]
ADD MINING MODEL [ZakaznikPohlavi-Tree]
(
	[IDZak],
	[Pohlavi] PREDICT,
	[Vek],
	[AutoTyp]
) USING Microsoft_Decision_Trees
```

V ukázce kódu jazyka DMX definujeme model rozhodovacího stromu, který předpovídá pohlaví v závislosti na věku osoby a typu auta, které osoba řídí.

Jak již bylo uvedeno výše, né každý algoritmus podporuje jakýkoli typ obsahu. Například v algoritmech Naiví Bayesova metoda a Asociační pravidla nejsou podporovány obsahy spojitého typu. Z tohoto důvodu jsme v předcházející části diskretizovali věk pomocí parametru `DISCRETIZED`.

```
ALTER MINING STRUCTURE [Zakaznik2]
ADD MINING MODEL [ZakaznikPohlavi-Bayes]
(
	[IDZak],
	[Pohlavi] PREDICT,
	[VekDisc] AS [Vek],
	[AutoTyp]
) USING Microsoft_Naive_Bayes
```

V ukázce můžeme vidět použití diskretizovaného sloupce `VekDisc` s názvem `Vek` v naivní Bayesově metodě. Toto nastavení je výhodné, pokud chceme zachovat shodné pojmenování sloupců ve všech dolovacích modelech ve struktuře.

Využitím vnořených tabulek v dolovací struktuře sebou přináší i lehce odlišnou tvorbu dolovacího modelu.

```
ALTER MINING STRUCTURE [Zakaznik3]
ADD MINING MODEL [PredictPohlaviNested-Trees]
(
	[IDZak],
	[Pohlavi] PREDICT,
	[Vek],
	[Nakupy]
	(
		[Produkt],
```

```
            [Mnozstvi],
            [Vyprodej]
      )
) USING Microsoft_Decision_Trees
```

Ukázka vytvoření modelu s vnořenou tabulkou.

Obdobně jako u dolovací struktury lze i u dolovacího modelu vytvářet podmnožinu dat pomocí filtrů. Lze samozřejmě také filtrovat vnořené případy. Sloupce struktury, na které se ve filtru odkazujeme, nemusí být obsaženy v definici modelu.

```
ALTER MINING STRUCTURE [Zakaznik3]
ADD MINING MODEL [FilterByVek]
(
      [IDZak],
      [Pohlavi],
      [Vek],
      [AutoTyp] PREDICT
) USING Microsoft_Decision_Trees

WITH FILTER(Vek > 30)
```

Ukázka vytváří dolovací model, který předpovídá typ auta založený na věku a pohlaví, ale pouze u zákazníků starších 30let.

4.3 Data miningové metody v SQL Serveru 2008

V následujících podkapitolách budou popsány všechny dostupné data miningové algoritmy v SQL Serveru 2008 Enterprise Edition. Ta oproti Standard Edition obsahuje podporu Plug-in algoritmů, paralelního zpracování dolovacích modelů a pokročilé nastavení a ladění data miningových algoritmů. Obě tyto edice obsahují komplexní soubor základních data miningových algoritmů jako jsou Asociační pravidla, Rozhodovací stromy, Časové řady, Shlukovací metody, Naivní Bayesovu metodu a Neuronové sítě.

4.4 Asociační pravidla

Algoritmus Asociačních pravidel, který se v originální terminologii SQL Serveru 2008 nazývá Microsoft Association Algorithm se využívá pro hledání vztahů v datech a objevování různých souvislostí. Nejčastěji se využívá v maloobchodě při tzv. analýze nákupního košíku. Jedná se o hledání asociací v nakupovaném zboží a následné využití této znalosti. Kupříkladu při zkoumání nákupních transakcí v nejmenovaném hypermarketu analytici zjistili, že ve středu večer mají zvýšený odbyt dětských plen, lahvového piva a slaného pečiva. Začali se pídit, z jakého důvodu k tomuto jevu dochází. Později zjistili, že častými nakupujícími v tuto dobu jsou muži kolem 30 let, kteří dostali za úkol koupit dětské pleny, a když už byli v obchodě, koupili si k tomu ještě pivo, ke kterému neopomněli něco slaného. Z pohledu managementu hypermarketu je tato informace velmi cenná, lze ji totiž využít několika způsoby. Například rozmístěním požadovaného zboží za účelem delšího pobytu zákazníka v obchodě a tím pádem dalšího potencionálního nákupu zboží. Další možností je umístit například lahvové pivo a slané pečivo bezprostředně vedle sebe a zákazníka tímto pobídnout k nákupu obou těchto produktů. Těchto

souvislostí lze využít i například k vytvoření speciálních nabídek.

Z hlediska matematické statistiky se tento jev nazývá korelace. Ta může být pozitivní nebo negativní. Pozitivní korelace je charakterizována tím, že vysoká úroveň jedné proměnné bude doprovázena vysokou úrovní korelační proměnné. Naopak negativní korelace popisuje, že zvýšení jedné proměnné bude provázena nízkou úrovní korelační proměnné (6). I tato souvislost je potencionálně důležitá.

U hledání souvislostí z dat nás zajímá, v kolika případech je splněn předpoklad, a v kolika závěr, dále kolikrát je splněn současně předpoklad i závěr, a další všechny kombinace. Zajímá nás pravidlo (3):

$$Ant \implies Suc$$

Ant (antecedent neboli předpoklad) a *Suc* (sukcedent neboli závěr) jsou kombinacemi kategorií vstupních dat. Tyto kombinace lze zapsat pomocí kontingenční tabulky:

	Suc	$\neg Suc$	$\sum$
Ant	a	b	r
$\neg Ant$	c	d	s
$\sum$	k	l	n

Tabulka 4.3- Kontingenční (čtyřpolní) tabulka

Základními charakteristikami asociačních pravidel dle Rakeshe Agrawala (11) jsou podpora a spolehlivost.

Podpora (Support) vyjadřuje počet objektů splňující předpoklad i závěr:

$$P\,(Ant \wedge Suc) = \frac{a}{a+b+c+d}$$

Spolehlivost (Confidence, v SQL Serveru 2008 označováno jako Probability) vyjadřuje podmíněnou pravděpodobnost závěru, pokud je předpoklad platný:

$$P\,(Suc \mid Ant) = \frac{a}{a+b}$$

Algoritmus Asociačních pravidel (Microsoft Association Algorithm) je realizován pomocí Apriori algoritmu. Ten postupně prochází případy (cases) a jejich položky. Asociované položky jsou následně seskupeny do množin, jejichž minimální podpora je nastavena parametrem MINIMUM_SUPPORT. Této množině se říká frekventovaná množina položek (frequent itemsets). Jednotlivé položky obsahují hodnoty atributů. Například pokud vzorek dat obsahuje atribut pohlaví, jednou položkou bude Pohlaví = muž. Součástí položek je i velikost (size), která vyznačuje počet hodnot v položce. Například položka {Pohlaví = muž, Věk = 30-35, TypAuta = Sedan} má velikost (size) rovno 3. V první iteraci algoritmus nalezne všechny položky s velikostí 1, následně provádí další iteraci pro položky s velikostí 2 atd. Na konci každé iterace algoritmus porovná soubor položek s podmínkou MINIMUM_SUPPORT a vybere pouze položky, které ji splňují. Iterace se postupně opakují pro velikost položek 3,4,5 atd., dokud nepřestanou položky splňovat kritérium MINIMUM_SUPPORT. Dalším krokem asociačního algoritmu je vytváření asociačních pravidel. Prohledávají se položky například Muž → Sedan nebo Sedan → Muž a hledají se pravidla s vysokou korelací. Pro každé z těchto pravidel

algoritmus počítá spolehlivost (probability), která se dá následně omezit parametrem MINIMUM_PROBABILITY. Tento parametr je třeba volit obezřetně, u řídkých dat lze nastavit MINIMUM_PROBABILITY na 50–10 procent a lze získat rozumné pravidla. V hustém souboru dat je doporučeno nastavit parametr na 40–50 procent, jinak se můžou vyskytnout i protichůdná pravidla jako například {Vysoké IQ → Pohlaví Muž} a {Vysoké IQ → Pohlaví Žena}. Další vlastností pravidla {A → B} je váha (importance), která se počítá jako poměr spolehlivosti {B | A} a spolehlivosti {B | ¬A}, který je následně normalizován pomocí logaritmické škály.

4.4.1 Parametry algoritmu Asociačních pravidel

Jak už bylo uvedeno v kapitole, algoritmus Asociačních pravidel je velmi citlivý na jeho vlastní nastavení. Z tohoto důvodu si zde uvedeme všechny parametry, jakými lze algoritmus nastavit pro co nejoptimálnější výkon a nejlepší výsledky.

- MINIMUM_SUPPORT

  Tento parametr definuje minimální podporu, kterou musí položka splnit, aby se stala součásti množiny frekventovaných položek (frequent itemsets). Pokud je její hodnota v rozmezí 0 až 1, je minimální podpora nastavena procentem. Například hodnota 0,25 vyznačuje, že položka se musí vyskytovat nejméně u 25 % záznamů. Pokud je hodnota minimální podpory vyšší než 1, jedná se o absolutní počet případů, které musí množina obsahovat. Jestliže není tento parametr uživatelem nastaven, výchozí hodnota je 0,03.

- MAXIMUM_SUPPORT

  Specifikuje maximální podporu položek, které jsou obsaženy ve frekventované množině položek. Tento parametr slouží k odfiltrování položek, které jsou příliš časté, a proto nemají tak vysoký význam. Pokud je hodnota parametru v rozmezí 0 až 1, je maximální podpora nastavena procentem. V opačném případě se jedná o absolutní počet případů, které může množina obsahovat. Výchozí hodnota je nastavena na 1, což znamená, že maximální podpora je 100%, proto budou použity všechny případy.

- MINIMUM_PROBABILITY

  Parametr minimální podpory specifikuje dolní hranici podpory asociačních pravidel. Jedná se o prahovou hodnotu. Výchozí hodnota algoritmu je nastavena na 0,4.

- MINIMUM_IMPORTACE

  Tento parametr nastavuje dolní hranici váhy (importance). Pravidla, která mají tuto hodnotu nižší, budou odfiltrována.

- MAXIMUM_ITEMSET_SIZE

  Specifikuje maximální velikost položky, která vyznačuje počet hodnot v položce. Snížením maximální velikosti lze snížit dobu potřebnou pro zpracování, algoritmus nebude provádět další iterace po dosažení této maximální velikosti. Výchozí hodnota je nastavena na 0, což znamená, že neexistuje žádné omezení na velikost položky.

- MINIMUM_ITEMSET_SIZE

Opačný případ předchozího parametru. V některých případech uživatele nezajímají menší položky, ale například položky, které mají počet hodnot v položce větší než 4. Tento parametr nesnižuje dobu zpracování, protože algoritmus musí postupně projít položky od velikosti 1 krok za krokem až do maximální velikosti.

- MAXIMUM_ITEMSET_COUNT

  Parametr určuje maximální počet položek množiny. Pokud není tento parametr nastaven, algoritmus vygeneruje všechny možné položky, které prošly podmínkou minimální podpory. Tento parametr zabraňuje vytváření příliš rozsáhlých frekventovaných množin položek.

- OPTIMIZED_PREDICTION_COUNT

  Tento parametr definuje počet položek, které budou uloženy do mezipaměti pro optimalizaci predikcí. Výchozí hodnota parametru je 0, což znamená, že algoritmus bude generovat tolik predikcí, kolik bylo požadováno v dotazu. Nastavením parametru na nenulovou hodnotu bude dotaz vracet nejvýše tolik predikcí, kolik je tato hodnota. Tímto nastavením lze zvýšit výkon predikčního dotazu.

- AUTODETECT_MINIMUM_SUPPORT

  Parametr představuje citlivost algoritmu, která se používá k automatické detekci minimální podpory. Nastavením této hodnoty na 1,0 způsobí, že algoritmus automaticky rozpozná nejmenší přiměřenou hodnotu minimální podpory. Jestliže je parametr nastaven na hodnotu 0, automatická detekce je vypnuta a používá se skutečná hodnota minimální podpory. Tento parametr je využíván pouze, pokud je MINIMUM_SUPPORT nastaveno na 0.

4.4.2 Asociační pravidla v jazyku DMX

Vytváření dolovací struktury a dolovacího modelu, který využívá algoritmu asociačních pravidel pomocí jazyka DMX.

```
CREATE MINING STRUCTURE Prodeje
(
[ID Objednavky] TEXT KEY,
Produkty TABLE
      (
            Produkt TEXT KEY,
            Kategorie TEXT DISCRETE
      )
)
```

V ukázce kódu jde vidět vytvoření nové dolovací struktury, která ještě není vázána na žádny algoritmus.

```
ALTER MINING STRUCTURE Prodeje
```

```
ADD MINING MODEL Doporuceni
(
	[ID Objednavky],
	Produkty PREDICT
	(
		Produkt
	)
) USING Microsoft_Association_Rules(MINIMUM_SUPPORT = 0.2,
MINIMUM_PROBABILITY = 0.4)
```

Nyní jsme vytvořili nový dolovací model, který používá algoritmu Asociačních pravidel, který má pomocí parametrů nastavenou minimální podporu na 20% a spolehlivost na 40%. Výstupem modelu budou doporučení pro produkty, na základě předcházejících nákupů.

V této podkapitole jsme popsali algoritmus asociačních pravidel, jeho základní charakteristiky a parametry. Následovala ukázka tvorby dolovacího modelu. V následující podkapitole se budeme věnovat shlukovacím metodám.

4.5 Shlukovací metody

Dalším dolovacím algoritmem v SQL serveru 2008 je shlukování. Tato metoda se používá k analýze vícerozměrných dat k nalezení podobnosti mezi těmito daty. Jedná se o proces organizování obdobných objektů do skupin, které se označují jako tzv. shluky (clusters). Jednou z nejvýznamnějších odlišností ve shlukovacím algoritmu je způsob jakým algoritmus rozhodne o zařazení případů do shluků. Prvním způsobem je tzv. hrubé shlukování (hard clustering), při kterém se vytvářejí disjunktivní shluky. Prvek je disjunktivní, pokud je spojen právě s jedním shlukem. Druhý způsob je jemné shlukování (soft clustering algorithms), které tvoří překrývající se shluky. Takovýto shluk je tvořen z prvků, které mohou náležet více shlukům současně.

V SQL Serveru 2008 existují dvě shlukovací metody, první je algoritmus shlukování v originální terminologii nazvaný Microsoft Clustering Algorithm. Druhým je sekvenční shlukování nazvaný Microsoft Sequence Clustering Algorithm, který si popíšeme níže.

Shlukování v SQL Serveru 2008 se provádí pomocí dvou metod:

- **EM shlukování** – Metoda EM (Expectation Maximization) shlukování patří do kategorie měkkého shlukování. Algoritmus pomocí vstupního shlukovacího modelu iterativně ověřuje pravděpodobnost, že daný bod leží v určitém shluku. Výpočet končí ve chvíli, kdy pravděpodobnostní model co nejvíce odpovídá skutečným datům. Pokud se za běhu algoritmu vyskytnou prázdné shluky, popř. shluky, které obsahují menší počet bodů, než je nastavena minimální hodnota, dochází k přesunu těchto shluků a EM algoritmus je následně spuštěn znovu. Výstupem této metody je souhrn pravděpodobností pro výskyt bodu v jednotlivých shlucích. Jelikož se jedná o jemné shlukování, každý bod může být obsažen i ve více shlucích, ale s různou pravděpodobností výskytu. V SQL Serveru 2008 jsou obsaženy dva způsoby EM shlukování. Prvním je škálovatelné EM shlukování, které načte prvních 50tis záznamů a provádí analýzu. V případě, že je úspěšná, použije se tento model i na další data.

Výhodou je rychlejší zpracování. Druhým způsobem je neškálovatelné EM shlukování, které provádí analýzu až po načtení všech záznamů, což může přinést zvýšení přesnosti, ale také vyšší nároky na paměť.

- **Metoda K-průměru (K-means)** – Metoda K-průměrů shlukování patří do kategorie hrubého shlukování. Algoritmus na základě vstupních dat začleňuje jednotlivé body do předem určeného počtu shluků podle vzdálenosti od centrálního bodu každého shluku. Každý body je přiřazen právě k jednomu shluku. Přiřazování je prováděno pomocí euklidovské vzdálenosti bodu od centrálního bodu shluku. Následně je pro každý shluk vypočítáno nové umístění centrálního bodu a výpočet pro přiřazení se opakuje. Pokud je bod s jiného shluku po přesunutí centrálního bodu umístěn blíže tomuto shluku, je přiřazen do bližšího shluku. Tento postup se opakuje, dokud není dosaženo výsledného rozmístění. Metoda K-průměrů obsahuje obdobně jako EM shlukování dva typy. Škálovatelná metoda K-průměrů načte prvních 50tis. záznamů a dále pokračuje ve výpočtu pouze v případě nutnosti. Neškálovatelná metoda K-průměrů načítá všechny záznamy a následně provádí shlukování.

4.5.1 Parametry algoritmu shlukování

Pomocí parametrů lze nastavit chování shlukovacího algoritmu (Microsoft Clustering Algorithm). Výchozí nastavení sice zpracuje většinu situací, ale za určitých podmínek lze dosáhnout lepších výsledků díky parametrům popsaných v této části (7).

- CLUSTERING_METHOD

  Tento parametr určuje metodu, pomocí které se bude provádět začlenění bodů do jednotlivých shluků. Možnosti parametru jsou:

  1 – Škálovatelná metoda EM shlukování
  2 – Neškálovatelná metoda EM shlukování
  3 – Škálovatelná metoda K-průměrů shlukování
  4 – Neškálovatelná metoda K-průměrů shlukování

  Výchozí hodnotou tohoto parametru je 1.

- CLUSTER_COUNT

  Specifikuje počet hledaných shluků pro určitý problém. Pokud algoritmus nenalezne požadovaný počet shluků z důvodu například velikosti vstupních dat, vygeneruje maximální nalezený počet. Jestliže vstupní data budou například záznamy o nákupech filmů, je vhodné neshlukovat konkrétní filmy, ale například jejich žánr. Tímto lze docílit menšího počtu shluků. Nastavení parametru na hodnotu 0 způsobí, že algoritmus heuristicky hledá ideální počet shluků. Výchozí hodnotou parametru je hodnota 10.

- MINIMUM_CLUSTER_CASES

  Nastavením tohoto parametru lze určit, kdy algoritmus považuje shluk za prázdný. Takto označený shluk se vyřadí a provádí se nový výpočet. Zavedením tohoto parametru je vhodné v případě, že je potřeba mít kupříkladu shluk menší 10. Výchozí hodnota je 1.

- MODELLING_CARDINALITY

  Tento parametr určuje počet vzorových modelů pro shlukování. Snížením tohoto počtu lze zvýšit výkon a rychlost výpočtu, ale tato volba přináší riziko nepřesnosti algoritmu z důvodu zahození vhodného vzorového modelu. Výchozí hodnota parametru je 10.

- STOPPING_TOLERANCE

  Parametr specifikuje hodnotu, která využívá pro určení dosažení konvergence a následného ukončení shlukování. Nastavená hodnota se kontroluje při každé iteraci algoritmu. Zvyšování této hodnoty bude mít za následek přesnější shlukování, ale delší zpracování, snižování zase přinese zkrácení doby shlukování s menší přesností nalezených shluků. Výchozí hodnota parametru je 10.

- SAMPLE_SIZE

  Tento parametr udává počet případů, používaných v jednotlivých krocích metody. Využívá se pouze při nastavení parametru CLUSTERING_METHOD na některou ze škálovatelných metod. Nastavení hodnoty na 0 způsobí, že se načtou všechna vstupní data a budou analyzována jedním průchodem. Tato volba může při rozsáhlých vstupních datech způsobit problémy s pamětí a výkonem. Výchozí hodnota je nastavena na 50 000.

- CLUSTER_SEED

  Parametr určuje náhodné číslo jádra, které se využívá k náhodnému generování shluků. Změnou hodnoty tohoto parametru lze změnit způsob vytváření počátečních shluků. Následně lze porovnat modely vybudované při různém nastavení. Pokud se výsledné shluky výrazně nemělní od vytvořeného modelu, lze jej následně považovat za stabilní. Výchozí hodnota je 0.

- MAXIMUM_INPUT_ATTRIBUTES

  Tento parametr určuje, jaké množství vstupních atributů bude algoritmu poskytnuto pro analýzu. Pokud existuje více atributů, než je uvedená hodnota, budou automaticky vybrány pouze nejvhodnější atributy. Ostatní atributy budou algoritmem ignorovány. Tento parametr je zaveden z důvodu časové náročnosti při analýze velkého množství atributů. Pokud je parametr nastaven na 0, není určeno žádné omezení. Výchozí hodnota je nastavena 255 atributů.

- MAXIMUM_STATES

  Parametr určuje maximální počet diskrétních hodnot, jakých může atribut nabývat. Jestliže bude atribut obsahovat větší množství hodnot, budou vybrány pouze ty nejčastější, ostatní hodnoty algoritmus ignoruje. Tento parametr je zaváděn z důvodu vlivu velkého množství hodnot na výkon a paměť. Výchozí hodnota je nastavena na 100.

V této podkapitole jsme si popsali první algoritmus shlukování a jeho dvě metody pro vyhledání shluků. V následující podkapitole bude vysvětlen algoritmus sekvenčního shlukování.

4.6 Sekvenční shlukování

Metoda sekvenčního shlukování (Microsoft Sequence Clustering Algorithm) se využívá k analýze dat, které obsahují určité sekvence kroků. Těmito sekvencemi se rozumí řada diskrétních událostí. Obvykle je počet těchto událostí konečný. Například sekvence kroků zákazníka při nákupu zboží v internetovém obchodě. Algoritmus následně analyzuje data pomocí těchto sekvencí a hledá určité podobnosti. Tyto informace můžou poskytnout přesnější pohled na chování zákazníka.

Algoritmus sekvenčního shlukování využívá metodu EM shlukování a analýzu Markovských řetězců. Markovský řetězec označuje náhodný (stochastický) proces s diskrétní množinou stavů. Proces je definován dvěma parametry: pravděpodobnostním vektorem a přechodovou maticí. Algoritmus nejprve analyzuje pravděpodobnosti přechodů a následně porovná vzdálenosti od již existujících sekvencí. Vhodné sekvence jsou pak použity jako vstup pro metodu EM shlukování. Shlukování sleduje dva druhy atributů – sekvenční a nesekvenční. Každý shluk obsahuje Markovský řetězec, který obsahuje množinu všech možných cest a matici, která popisuje sekvenci přechodů stavů a pravděpodobností. Dále je možné přidat i nesekvenční atributy, které připojí pomocí klasického shlukování. Tento postup vede k vytvoření velkého množství shluků, které by následně byly velmi obtížně interpretovány. Z tohoto důvodu algoritmus přistupuje k tzv. rozkladu shluků (cluster decomposition), při kterém se oddělují shluky, které obsahují sekvence, od shluků, které obsahují nesekvenční atributy.

4.6.1 Parametry sekvenčního shlukování

Algoritmus sekvenčního shlukování lze také parametrizovat. Úpravou těchto parametrů lze doladit používání tohoto algoritmu.

- CLUSTER_COUNT

  Obdobně, jako u shlukovacího algoritmu tento parametr určuje počet shluků, které mají být vytvořeny. Jestliže je hodnota nastavena na 0, algoritmus automaticky určí vhodný počet shluků pro prediktivní účely. Výchozí hodnota je 0.

- MINIMUM_SUPPORT

  Tento parametr určuje minimální počet případů, které jsou potřeba k vytvoření shluku. Jestliže je počet případů menší než hodnota parametru, považuje se shluk za prázdný a je vyřazen. Výchozí hodnota je nastavena na 10 případů

- MAXIMUM_STATES

  Definice tohoto parametru je stejná jako u shlukovacího algoritmu popsaného v předcházející podkapitole. Jedná se o maximální počet diskrétních hodnot atributů. Výchozí hodnota je 100.

- MAXIMUM_SEQUENCE_STATES

  Parametr určuje maximální počet stavů sekvenčního atributu. Výchozí hodnota je záměrně nastavena na 64. Předpokládejme, že máme M stavů. Každý vytvořený shluk obsahuje matici M*M. Doba zpracování je poté úměrná M^2. Jestliže je počet stavů M

příliš vysoký, prodlužuje se velmi doba zpracování tohoto modelu. Je doporučeno nenastavovat hodnotu parametru vyšší než 100. Pokud by například webová stránka obsahovala větší počet kroků, je vhodné přistoupit ke kategorizaci těchto kroků.

4.6.2 Sekvenční shlukování v jazyku DMX

V následující ukázce kódu v jazyku DMX bude tvorba dolovacího modelu pomocí algoritmu sekvenčního shlukování.

```
CREATE MINING MODEL WebSekvence
(
	ZakaznikID TEXT KEY,
	GeoLokace TEXT DISCRETE,
	PathLog TABLE PREDICT
	(
		SekvenceID LONG KEY SEQUENCE,
		URLKategorie TEXT DISCRETE PREDICT
	)
)
USING Microsoft_Sequence_Clustering(CLUSTER_COUNT = 4)
```

Vytvořili jsme dolovací model `WebSekvence`, který obsahuje vnořenou tabulku s klíčem sekvence `SekvenceID` a URL kategorie.

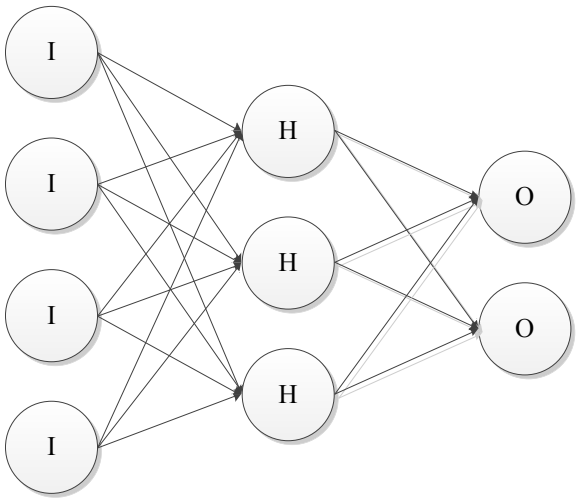
V této podkapitole jsme si popsali druhou shlukovací metodu, která využívá k analýze sekvence kroků.

4.7 Neuronové sítě

V originální terminologii SQL Serveru 2008 se tento algoritmus nazývá Microsoft Neural Network Algorithm. Analýza pomocí neuronových sítí nevychází z žádného statistického rozdělení, ale pracuje obdobně jako lidský mozek na principu rozpoznávání vzorů a minimalizování chyb. Lze si to představit jako přijímání informací a následné ponaučení z každé zkušenosti. Před započetím vlastního procesu se údaje rozdělí do trénovací a testovací množiny. Ta je následně iteračně zpracovávána systémem a porovnávána se skutečnými hodnotami. Je změřena chyba, na základě které se upravují původní váhy. Po dosažení předem určené minimální chyby proces končí. Neuronová síť je tvořena uzly uspořádanými do vrstev. Schéma neuronové sítě tzv. perceptron s jednou skrytou vrstvou obsahuje tři typy uzlů:

- I – vstupní uzly
- H – uzly skryté vrsty
- O – výstupní uzly

SQL Server 2008 využívá model mnohovrstvého perceptronu (viz. Obrázek níže). Jednotlivé neurony v jedné vrstvě mezi sebou nejsou propojeny, ale jsou propojeny se všemi neurony sousední vrstvy. Skrytá vrstva zajišťuje propojení mezi vstupní a výstupní vrstvou. Algoritmus podporuje predikci spojitých atributů, ale jako nejvhodnější je využití pouze diskrétních nebo diskretizovaných atributů.



Obrázek 4.2 - Schéma neuronové sítě

4.7.1 Parametry algoritmu Neuronové sítě

Běh algoritmu Neuronové sítě lze nastavit pomocí následujících parametrů.

- MAXIMUM_INPUT_ATTRIBUTES

  Parametr určuje maximální počet vstupních atributů, které mohou být poskytnuty algoritmu pro analýzu. Pokud je počet vstupních atributů vyšší než zadaná hodnota, algoritmus vybere nejvýznamnější a zbytek ignoruje. Nastavením parametru na hodnotu 0 je omezení počtu atributů vypnuto. Výchozí hodnota parametru je nastavena na 255.

- MAXIMUM_OUTPUT_ATTRIBUTES

  Parametr určuje maximální počet výstupních atributů analýzy. Pokud je počet výstupních atributů vyšší než zadaná hodnota, algoritmus vybere nejvýznamnější a zbytek ignoruje. Nastavením parametru na hodnotu 0 je omezení počtu atributů vypnuto. Výchozí hodnota parametru je nastavena na 255.

- MAXIMUM_STATES

  Definice tohoto parametru je stejná jako u shlukovacího algoritmu popsaného v předcházející podkapitole. Jedná se o maximální počet diskrétních hodnot atributů. Výchozí hodnota je 100.

- HOLDOUT_PERCENTAGE

  Parametr definuje procentuální podíl udržovaných dat, které se využívají k vyhodnocování přesnosti tréninku. Výchozí hodnota parametru je nastavena na 0,1.

- HOLDOUT_SEED

  Parametr určuje číslo, které je použito jako jádro generátoru pro náhodné rozpoznání pozastavovacích dat. Pokud je hodnota parametru nastavena na 0, generuje algoritmus jádro založené na názvu dolovacího modelu. Výchozí hodnota tohoto parametru je nastavena na 0.

- HIDDEN_NODE_RATIO

  Tento parametr se používá k nastavení počtu neuronů skryté vrstvy. Pokud je parametr nastaven například na hodnotu 2, množství neuronů ve skryté vrstvě se vypočítá vzorcem: 2 * sqrt(vstupní neurony * výstupní neurony). Výchozí hodnota tohoto parametru je 4.

- SAMPLE_SIZE

  Určuje maximální počet případů použitých pro trénování modelu. Výchozí hodnota je 10000.

4.8 Naivní Bayesův algoritmus

Naivní Bayesův algoritmus vychází z Bayesovy věty o podmíněných pravděpodobnostech. Toto tvrzení ukazuje, jak jedna podmíněná pravděpodobnost závisí na její inverzi. Přestože se jedná o metody pravděpodobnostní, využívají se i v dolování dat. Bayesů vztah pro výpočet podmíněné pravděpodobnosti. (3)

$$P\,(H \mid E) = \frac{P\,(E \mid H)\,P\,(H)}{P\,(E)}$$

Jednotlivé zastoupení hypotéz, je vyjádřeno tzv. apriorní pravděpodobností P(H). Následnou změnu pravděpodobnosti hypotézy v případě, že nastane událost E pak vyjadřuje podmíněná pravděpodobnost P(E | H). Pravděpodobnost evidence (pozorování) je vyjádřena veličinou P(E). Základním předpokladem naivního bayesovského klasifikátoru je předpoklad, že při platnosti hypotézy E jsou evidence H podmíněně nezávislé. Nejčastěji se Bayesovská klasifikace využívá např. při zjišťování pravděpodobnosti bonity klientů bank, filtrování spamu z E-mailových schránek, klasifikaci dokumentů apod. Microsoft Naive Bayes algorithm je určen pro rychlé vyhledávání vztahu mezi vstupními a predikovanými sloupci. Výsledkem je množina všech možných stavů predikovaného sloupce. Algoritmus podporuje predikci pouze diskrétních nebo diskretizovaných atributů.

4.8.1 Parametry naivní Bayesova algoritmu

- MAXIMUM_INPUT_ATTRIBUTES

  Parametr určuje maximální počet vstupních atributů, které budou analyzovány. Pokud bude na vstupu vyšší počet atributů, algoritmus vybere nejvýznamnější a zbytek ignoruje. Nastavení parametru na 0 způsobí, že algoritmus zanalyzuje všechny vstupní atributy. Výchozí hodnota je nastavena na 255.

- MAXIMUM_OUTPUT_ATTRIBUTES

  Parametr určuje maximální počet výstupních atributů. Nastavení parametru na 0 způsobí, že algoritmus zanalyzuje všechny výstupní atributy. Výchozí hodnota je nastavena na 255.

- MAXIMUM_STATES

  Tento parametr určuje maximální podporovaný počet diskrétních stavů atributu. Jestliže

je počet stavů pro daný atribut vyšší než hodnota tohoto parametru, použije algoritmus pro tento atribut nejvíce zastoupené stavy a ostatní ignoruje. Tento parametr je užitečný u atributů s vysokou mohutností například PSČ. Nastavením atributu na 0 zajistí, že algoritmus vezme v potaz všechny stavy atributu. Výchozí hodnota je nastavena na 100 stavů.

- MINIMUM_DEPENDENCY_PROBABILITY

  Parametr specifikuje minimální pravděpodobnost závislosti mezi vstupními a výstupními atributy. Tato hodnota je používána k nastavování limitů velikosti obsahu generovaným tímto algoritmem. Tato vlastnost nabývá hodnot od 0 do 1. Zvyšování hodnoty parametru se snižuje počet atributů v modelu. Výchozí hodnota je nastavena na 0,5.

4.9 Rozhodovací stromy

V původní terminologii se tento algoritmus nazývá Microsoft Decision Trees Algorithm. Diagram rozhodovacího stromu odhaluje závislosti a vyhledává specifické vlastnosti, které následně slouží k sestavení predikčního modelu rozhodování na jednotlivých úrovních hierarchické struktury. Tento rozhodovací algoritmus je využíván i v jiných vědních oborech například biologii. Zjednodušeně se jedná o rozdělování jednotlivých členů do kategorií na základě jejich parametrů.

K tvorbě rozhodovacího stromu se nejčastěji používá metoda - *rozděl a panuj*. Analyzovaná data se postupně rozdělují na menší a menší podmnožiny způsobe, aby v těchto podmnožinách převládaly příklady jedné třídy (3). Tento algoritmus se nazývá „top down induction of decision trees“ (TDIDT) a zde jsou jeho základní kroky:

1. Vyber jeden atribut, která bude kořen dílčího stromu.
2. Dalším krokem je rozdělení dat v tomto uzlu na podmnožiny dle hodnoty zvoleného atributu a poté přidej uzel pro každou podmnožinu.
3. Existuje-li uzel, pro který nepatří všechna data do shodné třídy, opakuj bod 1 pro tento uzel, jinak ukonči práci.

V SQL Serveru 2008 lze vytvářet rozhodovací stromy z diskrétních ale i spojitých atributů. Algoritmus vytváří rozhodovací strom rekurzivně. Takto mohou být vytvořeny poměrně veliké stromy, což nemá přímý vztah na kvalitu předpovědí. Tomuto problému algoritmus předchází díky tzv. před-prořezáváním stromu. Růst stromu je řízen pomoci Baysovského skóre, které se vyhýbá dalšímu štěpení, tam kde není dostatek informací. Toto je kontrolováno pomocí parametru COMPLEXITY_PENALTY, který může nabývat hodnot od 0 do 1. Čím vyšší číslo tím menší bude výsledný strom. Růst stromu je také řízen parametrem MINIMUM_SUPPORT, kterým lze nastavit minimální podporu atributu pro rozdělení uzlu.

4.9.1 Parametry rozhodovacích stromů

Funkci algoritmu rozhodovacích stromů je možné ovlivnit těmito parametry:

- COMPLEXITY_PENALTY

Tento parametr kontroluje růst rozhodovacího stromu. Nízká hodnota zvyšuje počet štěpů, vysoká jej naopak snižuje. Je doporučeno vytvořit více stromů s různým nastavením a následně určit nejpřesnější. Výchozí hodnota je nastavena podle počtu atributů konkrétního modelu:

- Pro méně než 10 atributů je výchozí hodnota 0,5.

- Pro 10 až 99 atributů je výchozí hodnota 0,9.

- Pro 100 a více atributů výchozí hodnota 0,99.

- MINIMUM_SUPPORT

  Minimální podpora listových hodnot potřebných pro vytvoření potomka v rozhodovacím stromu. Tuto hodnotu je vhodné zvýšit, pokud je množina zpracovávaných dat příliš velká, vyhneme se tak přetrénování (overtraining). Výchozí hodnota je 10.

- SCORE_METHOD

  Tento parametr se používá k určení metody pro stanovení rozdělení skóre při růstu stromu. Existují tři metody:

  1 – Použití „Entropy“ skóre.

  2 – Algoritmus využije „Bayesian s K2 Prior“ metodu, která ke každému atributu přidává konstantu bez ohledu na úroveň stromu.

  3 – Metoda „Bayesian Dirichlet Equivalent with Uniform Prior“ je nastavena jako výchozí. Nastavuje každému uzlu váhu, která se počítá z podpory uzlu a úrovní ve stromu.

- SPLIT_METHOD

  Určuje metodu použitou pro následné určení tvaru stromu. K dispozici jsou následující možnosti.

  1 - Binární: Indikuje, že strom má být rozdělen binárním způsobem, nezávisle na číselné hodnotě atributu.

  2 - Kompletní: Indikuje, že strom může vytvářet tolik štěpů, kolik má atribut hodnot.

  3 - Obojí: Algoritmus pro analýze sám rozpozná, kterou z rozdělovacích metod je vhodné použít pro získání nejlepších výsledků.

  Výchozí hodnota je nastavena na 3.

- FORCE_REGRESSOR

  Parametr algoritmu nastaví použití určeného sloupce regresorem, bez ohledu na důležitost sloupců určenou algoritmem. Tento parametr se využívá pouze pro stromy predikující atributy se spojitou hodnotou.

- MAXIMUM_INPUT_ATTRIBUTES

  Parametr určuje maximální počet vstupních atributů, které mohou být poskytnuty

algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro vstupní atributy vyřadí. Výchozí hodnota je nastavena na 255.

- MAXIMUM_OUTPUT_ATTRIBUTES

  Parametr určuje maximální počet výstupních atributů, které mohou být poskytnuty algoritmu před aplikováním selekce rysů. Nastavení na 0 selekci rysů pro výstupní atributy vyřadí. Výchozí hodnota je nastavena na 255.

4.10 Časové řady

Algoritmus časových řad se v terminologii SQL Serveru 2008 nazývá Microsoft Time Series Algorithm. Tato metoda analyzuje proměnné jako například obrat, náklady, zisk atd. z pohledu časových řad. Na základě analýzy dat z minulosti lze nalézt určitá pravidla a na základě nich poté provádět predikce dané proměnné. Jedná se o regresi časových úseků, takže předpověď možného vývoje veličiny v sobě obsahuje i krátkodobé opakující se výkyvy (fluktuace). Například pokud budeme sledovat vývoj nezaměstnanosti pro následující rok na základě dat z minulých let, objeví se v kvalitní predikci i logické výkyvy (fluktuace), jako nárůst nezaměstnanosti po dokončení studia nebo pokles při sezónních pracích. Algoritmus časových řad v SQL Serveru 2008 obsahuje dva algoritmy. ARTxp algoritmus byl obsažen již v předcházející verzi SQL Server 2005. Druhý algoritmus byl přidán až do stávající verze 2008.

- **ARTxp** – jedná se v podstatě o algoritmus rozhodovacích stromů v kombinaci s automatickou regresí. Ta využívá lineární regrese k nalezení závislosti mezi dvěma proměnnými, kdy jedna z těchto proměnných je čas. Algoritmus nejprve provede transformaci dat. Při vstupu více dat umožňuje ARTxp algoritmus provádět tzv. křížené predikce. Ta umožňuje použití dvou oddělených příbuzných řad pro vytvoření jedné výsledné predikce. Například když prodejnost jednoho produktu ovlivňuje prodejnost druhého produktu.
- **ARIMA** – tento algoritmus se využívá k vytvoření dlouhodobých předpovědí. Funguje na principu pohyblivého průměru. Jinými slovy předpovídá hodnoty v sérii na základě rozdílu mezi skutečnými hodnotami a predikovanými. ARIMA model je popsán jako *ARIMA (p, d, q)*, kde *p* představuje počet podmínek automatické regrese, *d* počet rozdílů a *q* počet chyb při predikci.

4.10.1 Parametry algoritmu časových řad

Algoritmus časových řad lze ladit a nastavovat jeho chod pomocí následujících parametrů.

- MISSING_VALUE_SUBSTITUTION

  Tento parametr určuje, jakým způsobem budou ve vstupních datech vyplněny mezery. Výchozí nastavení tohoto parametru nepovoluje žádné mezery. Pokud se vyskytne alespoň jedna nevyplněná hodnota, algoritmus zhavaruje. Nabízí se otázka, proč není nastavena výchozí hodnota? Odpovědí je, že způsob vyplnění chybějících hodnot v časové řadě má podstatný vliv na výsledný model. Bezpečnější je analytikovi zobrazit chybné hlášení a donutit ho tuto nesrovnalost napravit. Jedním způsobem je doplnit data o chybějící údaje tak, aby neexistovaly žádné mezery. Druhou možností je nastavit tento

parametr na jednu z následujících hodnot:

`Previous` (předchozí) – Toto nastavení způsobí, že algoritmus použije hodnotu z předchozího časového intervalu, kdykoli najde mezeru ve vstupních datech. Tato volba je vhodná pro různé scénáře, například u konečné hodnoty skladu ve dnech, kdy se neobchodovalo a dá se předpokládat stejná hodnota z minulého dne.

`Mean` (střední) – Tato volba nahradí chybějící data průměrnou hodnotou celé série. Tohle nastavení není příliš vhodné, ale je užitečné v případě, kdy chcete nahradit mezeru v datech jakoukoli hodnotou, která příliš neovlivní celý výpočet.

Číslo – Zapsání číselné hodnoty způsobí, že algoritmus všechny mezery v datech nahradí touto hodnotou. Tato volba je vhodná v situacích, kdy je chybějící hodnota známá nebo obvyklá. Například, pokud chybějící hodnota vyznačuje, že se produkt neprodával v tomto časovém období, lze hodnotu nastavit na číslo 0.

- PERIODICITY_HINT

  Druhý velmi podstatný parametr, pomocí kterého lze nastavit periodicitu (pravidelnost) dat. Vzhledem k tomu, že správná periodicita dat může znamenat rozdíl mezi správným a špatným modelem, je vhodné ji nastavit, pokud je známá. Hodnota parametru se udává jako celé číslo, které znamená měrnou jednotku jeden měsíc ve formátu {n [, n]}. Například pokud bychom chtěli určit periodicitu příjmů firmy kvartálně a ročně, měl by tento parametr hodnotu {12, 3}. Výchozí hodnota parametru je 1.

- AUTO_DETECT_PERIODICITY

  Pomocí tohoto parametru lze nastavit, jak průbojný bude algoritmus při hledání periodicity v datech. Hodnota parametru se pohybuje mezi hodnotami 0 a 1. Pokud bude nastavena hodnota 0, bude algoritmu vyhledávat pouze nejsilnější periodicitu. V opačném případě, pokud je nastavena hodnota 1, algoritmus najde i sebemenší pravidelnost v datech, což se projeví na vyšší přesnosti modelu, ale také na delším čase zpracování. Výchozí hodnota je nastavena na 0,6.

- MINIMUM a MAXIMUM_SERIES_VALUE

  Tyto dva parametry umožňují určit rozmezí rozsahu hodnot pro platné predikce. Například pokud provádíme předpověď zásob určitého zboží na skladě a dosavadní průběh ukazuje, že se zásoba snižují, je možné, že by další průběh mohl naznačit postup až do záporných hodnot, což není u této proměnné možné. Nastavením parametru na číslo 0 docílíme, že záporné hodnoty budou odříznuty.

- FORECAST_METHOD

  Tímto parametrem lze nastavit, který algoritmus bude použit pro hledání predikcí. Oba tyto algoritmy jsou popsány výše. Výchozí hodnota je nastavena na `MIXED`, což naznačuje, že budou použity oba algoritmy a výsledkem bude směs predikcí obou. Další možné nastavení parametru jsou `ARIMA` a `ARTXP` (který simuluje chování algoritmu v SQL Serveru 2005).

- PREDICTION_SMOOTHING

Tento parametr nastavuje rovnováhu směsi mezi výše popsanými algoritmy pro hledání předpovědí. Parametr může nabývat hodnot od 0 do 1. Čím bližší bude hodnota číslu 0, tím více budou použity predikce z ARTxp algoritmu. Naopak hodnota blížící se číslu 1 znamená, že do výsledné směsi predikcí budou použity více výsledky ARIMA algoritmu. Tento parametr se využívá pouze, pokud je FORECAST_METHOD nastavena na `MIXED`.

- INSTABILITY_SENSITIVITY

  Tento parametr byl zaveden již v SQL Serveru 2005 z důsledku masivní zpětné vazby od zákazníků, z důvodu nestability ARTxp algoritmu, který při překročení určité hranice rozptylu ukončil predikci. Tento parametr dává možnost detekci nestability vypnout. Pokud je hodnota nastavena na 1, chová se algoritmus jako při původním nastavení v SQL Serveru 2005. Pokud je parametr nastaven na 0, je detekce nestability zcela vypnuta. Výchozí hodnota je 1. Parametr se používá pouze v případech, kdy je FORECAST_METHOD nastaven na `ARTXP`. Tento parametr lze modifikovat pouze ve verzi Enterprise.

- HISTORICAL_MODEL_COUNT a HISTORICAL_MODEL_GAP

  Parametr HISTORICAL_MODEL_GAP specifikuje časový interval mezi dvěma historickými časovými modely. Tato hodnota určuje počet časových jednotek v měsících. Výchozí hodnota je nastavena na 10. Parametr HISTORICAL_MODEL_COUNT představuje počet těchto historických modelů, které mají být vytvořeny.

- COMPLEXITY_PENALTY a MINIMUM_SUPPORT

  Tyto dva parametry se využívají jen zřídka. Používají se ve spojení s ARTxp algoritmem. Zvýšením COMPLEXITY_PENALTY se redukuje větvení rozhodovacího stromu a snižuje se tím růst stromu. MINIMUM_SUPPORT určuje minimální počet časových úseků potřebných k vytvoření větvení rozhodovacího stromu. Výchozí hodnota je nastavena na 10.

4.10.2 Časové řady v jazyku DMX

Ukázka vytvoření dolovací struktury a dolovacího modelu s použitím algoritmu časových řad v jazyku DMX.

```
CREATE MINING STRUCTURE [Prodej Vina]
(
      [Mesic] DATE KEY TIME,
      [Druh] TEXT KEY,
      [Kategorie] TEXT DISCRETE,
      [Prodeje] DOUBLE CONTINUOUS
)
```

Vytvoření dolovací struktury, která obsahuje časový klíč `Mesic`, dále druh vína, kategorii vína a prodejnost.

```
ALTER MINING STRUCTURE [Prodej Vina]
ADD MINING MODEL [Cervene Vino]
(
	[Mesic],
	[Druh],
	[Prodeje] PREDICT
) USING Microsoft_Time_Series
WITH DRILLTHROUGH,

FILTER([Kategorie] = 'Cervene')
```

Ukázka vytvoření dolovacího modelu, který je filtrován pouze na červené víno. Parametr `DRILLTHROUGH` se používá v okamžiku, kdy chceme zobrazit případy, které byly použity v tréninkovém modelu ve srovnání použití testovacího modelu.

V této podkapitole byly vysvětleny časové řady a jejich dva algoritmy ARTxp a ARIMA. Dále byly popsány všechny atributy, kterými lze časové řady ladit. Následovala ukázka použití v jazyku DMX.

5 Datová analýza

V kapitole jsem vycházel z této literatury: (4), (5), (6), (10)

5.1 Datový sklad

Datový sklad je strukturovaný repositář rozsáhlých dat, který umožňuje provádět efektivní analýzu a dotazování. Uložení dat je realizováno odděleně, aby jejich náročné zpracování nenarušovalo provoz samotného systému. Základními vstupními daty jsou systémové databáze, ale i archívy a jiné externí zdroje. Data jsou nahrávána jednorázově pomocí tzv. datových pump. Z toho důvodu nejsou zatěžovány lokální systémové databáze a je zajištěn vyšší výkon pro vyhodnocování dotazu.

5.2 OLAP

Analytické databáze jsou často označovány pojmem OLAP (Online Analytical Processing). Tento pojem zahrnuje struktury dat a analytické služby, které slouží pro analýzu objemných dat uložených obvykle v datových skladech.

5.2.1 Logická struktura

Pro vytvoření dimenzionální OLAP struktury je potřeba dvou druhů dat:

- **Dimenze** – tabulky, které obsahují logicky nebo organizačně uspořádaná data (Kategorie, Otázka, Uživatel, Kurz, Test). Tyto tabulky jsou řádově menší, než tabulky faktů a obsah se v nich tak často nemění.
- **Fakta** – tabulky obsahující atributy, které jsou hlavním důvodem evidence a jejíž hodnoty nás budou zajímat pro další analýzy a rozhodování (počet bodů, hodnocení, správnost odpovědi). Tabulky faktů obsahují velký objem dat a jsou to zpravidla největší tabulky databáze.

5.2.2 Modelování dat

V dimenzionální úrovni se nejčastěji využívají dva základní přístupy:

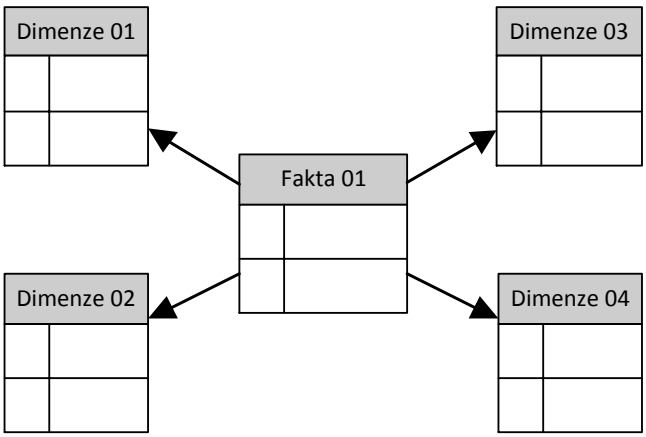
- **ROLAP** (Relační OLAP) – Relační online analytické zpracování dat získává data pro analýzy z relačního datového skladu. Používají se stávající databázové technologie. Využívá se relačních tabulek (fakt a dimenzí) organizovaných do hvězdicových schémat.
- **MOLAP** (Multidimenzionální OLAP) – Pro multidimenzionální online analytické zpracování se získají data z datového skladu, mechanismus MOLAP následně uloží analytická data ve vlastních strukturách a sumářích.

Dalšími variantami jsou:

- **HOLAP** (Hybridní OLAP) – Kombinace úložišť ROLAP a MOLAP, kde se využívá výhod jednotlivých typů, díky čemu se dají do značné míry eliminovat jejich nevýhody.
- **DOLAP** (Desktop OLAP) – Datový sklad vedený na klientském lokálním počítači. Veškeré analytické operace jsou následně prováděny nad touto lokální kostkou.

5.2.3 Hvězdicové schéma

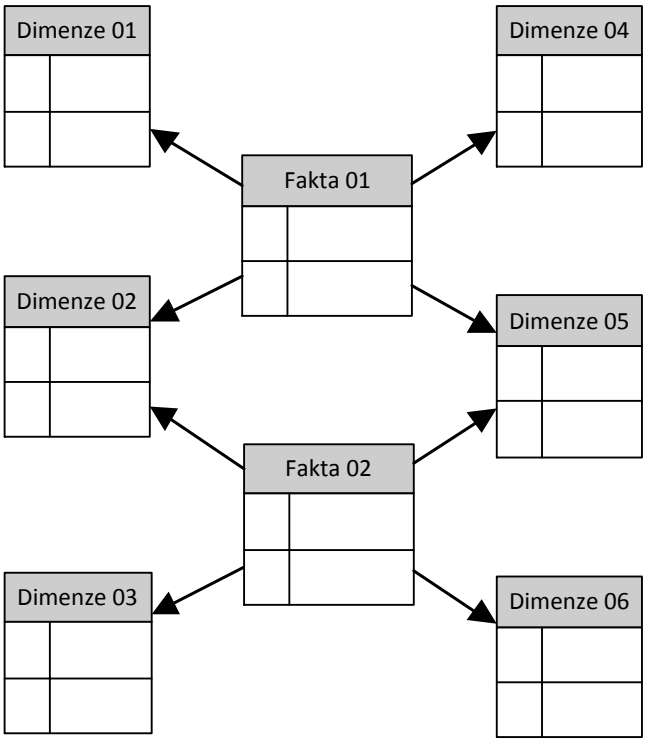
Nejčastěji se pro topologické uspořádání používá hvězdicové schéma (star schema). Dalším možným je schéma sněhové vločky (snowflake schema). Hvězdicové schéma je složeno z tabulky faktů, která obsahuje cizí klíče. Ty se vztahují k primárním klíčům v tabulkách dimenzí.



Obrázek 5.1 - Hvězdicové schéma

5.2.4 Schéma souhvězdí

Schéma souhvězdí je rozšířením hvězdicového schématu. Jednotlivé dimenze mohou být sdíleny mezi různými tabulkami faktů.



Obrázek 5.2 - Schéma souhvězdí

6 Data mining v systému eLogika

V kapitole jsem vycházel z této literatury: (3), (4), (5) (6), (7), (10), (12), (13)

V této kapitole budou popsány jednotlivé kroky, které bylo potřeba provést pro použití vybraných data miningových metod v systému eLogika.

Pro zdárnou aplikaci data miningových metod v systému jsem zvolil použití metodologie CRISP-DM, jejíž kroky jsou podrobně popsány v druhé kapitole. Proto i tato kapitola bude strukturována na základě postupů a kroků z této metodologie. Prvním krokem je porozumění problému a stanovení cílů a požadavků.

6.1 Stanovení cílů a požadavků

Prvotním impulzem myšlenky použití data miningových metod v e-learningovovém systému eLogika bylo využití potenciálu těchto algoritmů pro zlepšení kvality studia a pomoci tím studentům lépe zvládnout látku. Základním cílem tedy bylo získat ze systému jisté potenciálně užitečné informace o studentech a jejich průběhu studia a využít tyto znalosti.

Nejprve je potřeba dobře pochopit procesy, které v systému eLogika fungují a vytýčit si konkrétní entity, které budou data miningovými algoritmy analyzovány. Jak už bylo popsáno v první kapitole, eLogika je e-learningový systém, který je využíván jak studenty, tak vyučujícími. Základní jednotkou systému z pohledu studenta je kurz. Jedná se o předmět, který je vyučován v rámci školy. Každý kurz má nastaveny podmínky, které je třeba splnit pro zvládnutí vyučované látky a zvládnutí celého kurzu. V průběhu studia kurzu má student možnost ověřit si své znalosti buďto sám pomocí cvičných testů, nebo tyto znalosti budou ověřeny vyučujícím pomocí zápočtových testů. Na konci kurzu jsou studentovy znalosti ověřeny pomoci hlavních (zkouškových) testů. Výsledky ze všech těchto testů jsou uchovávány v systému eLogika a právě tyto informace budou vstupními daty pro následnou analýzu pomocí data miningových metod. Podrobněji budou tato data popsána v dalších podkapitolách.

6.2 Porozumění datům ze systému eLogika

V této kapitole se budeme věnovat podrobnému popisu analyzovaných dat systému eLogika. Podle metodologie CRISP-DM je potřeba provést tyto kroky: sběr analyzovaných dat, popis těchto dat, průzkum dat a ověřování kvality dat.

Jak již bylo popsáno v předcházející podkapitole, základními analyzovanými daty budou informace o všech provedených testech studenty. Veškeré tyto informace jsou systémem eLogika uchovávány v jeho databázích. Pro správné porozumění datům je potřeba znát strukturu testů. Každý test je složen z otázek obsažených v kategoriích, které jsou vloženy v jednotlivých kurzech. V systému eLogika je možné vytvořit několik druhů testů, ať už se jedná o testy vygenerované konkrétně pro určitého studenta, nebo testy na skupiny, kdy je vytvořeno několik verzí testů například A, B, C, D a ty jsou následně rozdány mezi studenty. Testy se dále rozdělují na online a vytištěné. Z tohoto důvodu se liší i sběr analyzovaných dat. Pokud student vyplňuje test přímo přes webové rozhraní systému eLogika, po dokončení je test uložen rovnou do databáze. Odlišná situace nastává, pokud se jedná o tištěný test. Po vyplnění a odevzdání

testu studentem jsou dvě možnosti, jakým bude výsledek uložen do systému.

První možností je ruční zavedení odpovědí studenta do systému. Tuto možnost provádí vyučující. Pokud se jedná o otázky, na které se odpovídá volbou a, b, c, d atd., vyučující zadá vyplněné odpovědi studenta. U druhého typu otázek tzv. formulářových vyučující zadá procentuální úspěšnost podle zhodnocení studentovy textové odpovědi. Po dokončení jednotlivých testů se informace nahrají do databáze.

Druhým způsobem uložení vyplněných tištěných testů do systému eLogika je použití modulu rozpoznávače tištěných testů. Studenti vyplní své odpovědi do tzv. odpovědníkového listu, který je součástí tištěného testu. Tyto odpovědníkové listy se následně hromadně skenují do počítače a zpracovávají pomocí doprovodného softwaru. Výsledky se poté nahrají do systému eLogika, který je uloží do databáze. Podrobněji tuto problematiku popisuje Bc. Tomáš Loskot ve své diplomové práci „Analýza pro systém eLogika“.

Data týkající se provedených testů studentů se nacházejí v databázi systému eLogika v těchto tabulkách a vazebních tabulkách: vygenerovany_test, vygenerovany_test_uzivatel, uzivatel, vygenerovana_otazka, vygenerovana_odpoved, odpovedi, otazky, vysledek, hodnoceni_otazka, hodnoceni_otazka_vysledek, hodnoceni_test, popř. doprovodné tabulky. Podrobně jsou tyto tabulky a jejich atributy popsány v návrhu databáze, která je součástí diplomové práce „Systém eLogika a e-learningová podpora výuky“, kterou zpracoval Bc. Vojtěch Hernas.

Z důvodu postupného vývoje systému eLogika a souběžného používání v ostrém provozu se do databáze v některých případech zavedla nekvalitní data. Z tohoto důvodu bylo potřeba provést ověření dat. Některé chyby v datech byly odhaleny a následně opraveny již při prvním náhledu do dat. Jiné vyšly najevo až při zpracování data miningových metod. Proto se v rámci metodologie CRISP-DM neustále vracíme ke kroku Příprava dat.

V následující podkapitole bude popsána příprava dat a tvorba datového skladu systému eLogika.

6.3 Příprava dat a tvorba datového skladu

V této podkapitole bude popsáno, jakým způsobem budou data uložena v datovém skladu systému eLogika. Dále bude vysvětleno plnění datového skladu pomocí ETL procesů vytvořených v SQL Serveru Integration Services.

Cílem této fáze metodologie CRIPS-DM je vytvoření datového souboru, v našem případě datového skladu, pro následnou analýzu pomocí data miningových metod. Nyní si popíšeme jednotlivé kroky k vytvoření datového skladu systému eLogika.

6.3.1 Fakta a dimenze

Prvním krokem tvorby datového skladu je rozvaha nad analyzovanými daty a hledání faktů a dimenzí. Tyto pojmy byly podrobně vysvětleny v předcházející kapitole. Pro zopakování tabulka faktů obsahuje atributy, které jsou v analýze hlavním důvodem evidence. Tabulka dimenze obsahuje logicky nebo organizačně strukturovaná data.

Jak již bylo uvedeno v přecházejících podkapitolách, základními analyzovanými daty budou informace o provedených testech studentů. Vygenerovaný test systémem eLogika se skládá z jednotlivých otázek. Každá otázka v testu má určité množství odpovědí. Pokud se na tuto

problematiku podívám z pohledu množit, tak hlavní množinou bude test. Jejími podmnožinami následně budou otázky a podmnožinou těchto otázek jsou odpovědi. Pro každou z těchto entit lze uchovávat její hodnoty, popř. dopočítat jiné. Tyto hodnoty pak budou předmětem analýzy pomocí data miningových metod. Z tohoto důvodů lze tyto entity považovat za fakta.

Nyní si popíšeme jednotlivé atributy tabulky faktů:

Tabulka faktů test – `fact_test`

Tabulka faktů test obsahuje data o veškerých testech provedených v systému eLogika. Každý individuální záznam v tabulce představuje jeden provedený test.

fact_test		
Název atributu	**Datový typ**	**Popis**
id_fact_test	int	(PK) Identifikátor tabulky faktů test
id_kurz_info	int	(FK) Identifikátor dimenze kurzů, Kurz pro který byl test vypracován
id_kurz	int	Identifikátor kurzu, Instance kurzu v určitém rozdělení akademického roku
id_test	int	(FK) Identifikátor dimenze test, Konkrétní test
id_vygenerovany_test	int	Identifikátor vygenerovaného testu
id_uzivatel	int	(FK) Identifikátor dimenze uživatel, Uživatel, který test vyplnil
id_tutor	int	(FK) Identifikátor dimenze uživatel, Tutor, který studenta vyučoval
id_date	int	(FK) Identifikátor dimenze data, Datum vypracování testu
id_time	int	(FK) Identifikátor dimenze času, Čas vypracování testu
skupina	varchar(10)	Varianta testu
id_trida	int	(FK) Identifikátor dimenze třída, Třída studenta
forma_studia	int	Identifikátor formy studia studenta
max_body_test	float	Maximální počet bodů za test
body_test	float	Získaný počet bodů za test
body_test_proc	float	Procentuální počet získaných bodů za test
uspesnost	varchar(10)	Úspěšnost studenta
id_zarazeni	int	Identifikátor zařazení testu (hlavní, cvičný, ukázkový)

zarazeni	varchar(50)	Zařazení testu (hlavní, cvičný, ukázkový)
pokus_zk	int	Pokus zkoušky studenta
poc_cv_test_ok	int	Počet správně vyplněných cvičných testů
poc_cv_test_x	int	Počet špatně vyplněných cvičných testů
id_date_prihl	int	(FK) Identifikátor dimenze data, Datum přihlášení na test
id_time_prihl	int	(FK) Identifikátor dimenze času, Čas přihlášení na test
prihl_poc_dni	int	Počet dnů přihlášení na test

Tabulka 6.1 - Atributy tabulky faktů test

Atribut `body_test_proc` se dopočítává vzorcem ((`body_test / max_body_test`) `* 100`). Atributy `poc_cv_test_ok` a `poc_cv_test_x` se dopočítávají způsobem, že ke každému záznamu se prochází všechny provedené testy nižšího nebo stejného data a hledají se pouze cvičné testy. Atribut `uspesnost` se vyplňuje podle atributu `body_test_proc` a může nabývat hodnot (0%-20%, 20%-40%, 40%-60%, 60%-80% nebo 80%-100%). Hodnota atributu `prihl_poc_dni` je rozdíl dní mezi datem přihlášení a datem vypracování testu.

Tabulka faktů otázka – `fact_otazka`

Tabulka faktů otázka obsahuje hodnoty o jednotlivých otázkách provedených v rámci testu.

fact_otazka		
Název atributu	**Datový typ**	**Popis**
id_fact_otazka	int	(PK) Identifikátor tabulky faktů otázka
id_kurz_info	int	(FK) Identifikátor dimenze kurzů, Kurz pro který byla otázka vypracována
id_kurz	int	Identifikátor kurzu, Instance kurzu v určitém rozdělení akademického roku
id_test	int	(FK) Identifikátor dimenze test, Test, ve kterém byla otázka vypracována
id_vygenerovany_test	int	Identifikátor vygenerovaného testu
id_uzivatel	int	(FK) Identifikátor dimenze uživatel, Uživatel, který na otázku odpověděl
id_kategorie	int	(FK) Identifikátor dimenze kategorie, Kategorie, do které otázka spadá
id_kategorie_puv	int	(FK) Identifikátor dimenze kategorie, Původní kategorie, do které otázka spadá

id_otazka	int	(FK) Identifikátor dimenze otázka, Konkrétní vypracovaná otázka
id_otazka_puv	int	(FK) Identifikátor dimenze otázka, Konkrétní vypracovaná původní otázka
id_tutor	int	(FK) Identifikátor dimenze uživatel, Tutor, který studenta vyučoval
id_date	int	(FK) Identifikátor dimenze data, Datum vypracování otázky
id_time	int	(FK) Identifikátor dimenze času, Čas vypracování otázky
skupina	varchar(10)	Varianta testu
typ_hodnoceni	int	Způsob hodnocení otázky
id_trida	int	(FK) Identifikátor dimenze třída, Třída studenta
forma_studia	int	Identifikátor formy studia studenta
max_body_otazka	float	Maximální počet bodů za otázku
body_otazka	float	Získaný počet bodů za otázku
body_otazka_proc	float	Procentuální počet získaných bodů za otázku
uspesnost	varchar(10)	Procentuální úspěšnost studenta za otázku
id_zarazeni	int	Identifikátor zařazení otázky (hlavní, cvičná, ukázková)
zarazeni	varchar(50)	Zařazení otázky (hlavní, cvičná, ukázková)
cas_zobrazeni	time(0)	Čas zobrazení otázky
cas_zobrazeni_sec	int	Čas zobrazení otázky v sekundách
pocet_zobrazeni	int	Počet zobrazení otázky
pokus_zk	int	Pokus zkoušky studenta
poc_cv_ot_kat_ok	int	Počet správně vyplněných cvičných otázek ve stejné kategorii
poc_cv_ot_kat_x	int	Počet špatně vyplněných cvičných otázek ve stejné kategorii

Tabulka 6.2 - Atributy tabulky faktů otázka

Atribut `uspesnost` nabývá stejných hodnot, jako u předcházející tabulky faktů, jen je vyplňován na základě atributu `body_otazka_proc`. Atributy `cas_zobrazeni` a `pocet_zobrazeni` jsou vyplněny pouze u testů vyplněných on-line. Jedná se o hodnoty

získané z průběhu on-line testu. Atributy `poc_cv_ot_kat_ok` a `poc_cv_ot_kat_x` se dopočítávají způsobem, že ke každému záznamu se prochází všechny provedené otázky nižšího nebo stejného data a hledají se pouze cvičné otázky ze stejné kategorie.

Tabulka faktů odpověď – `fact_odpoved`

Tabulka faktů odpověď obsahuje záznamy o zodpovězených odpovědích studentem. Každý jednotlivý záznam je jeden výskyt odpovědi určité otázky, určitého testu.

fact_odpoved		
Název atributu	**Datový typ**	**Popis**
id_fact_odpoved	int	(PK) Identifikátor tabulky faktů odpověď
id_kurz_info	int	(FK) Identifikátor dimenze kurzů, Kurz pro který byla otázka vypracována
id_kurz	int	Identifikátor kurzu, Instance kurzu v určitém rozdělení akademického roku
id_test	int	(FK) Identifikátor dimenze test, Test, ve kterém byla odpověď označena
id_vygenerovany_test	int	Identifikátor vygenerovaného testu
id_uzivatel	int	(FK) Identifikátor dimenze uživatel, Uživatel, který odpověď odpověděl
id_kategorie	int	(FK) Identifikátor dimenze kategorie, Kategorie, do které otázka spjatá s touto odpovědí
id_otazka	int	(FK) Identifikátor dimenze otázka, Otázka pod kterou spadá tato odpověď
id_odpoved	int	(FK) Identifikátor dimenze odpověď, Konkrétní zodpovězená odpověď
id_tutor	int	(FK) Identifikátor dimenze uživatel, Tutor, který studenta vyučoval
id_date	int	(FK) Identifikátor dimenze data, Datum zodpovězení odpovědi
id_time	int	(FK) Identifikátor dimenze času, Čas zodpovězení odpovědi
skupina	varchar(10)	Varianta testu
typ_hodnoceni	int	Způsob hodnocení otázky
id_trida	int	(FK) Identifikátor dimenze třída, Třída studenta

forma_studia	int	Identifikátor formy studia studenta
max_body_otazka	float	Maximální počet bodů za otázku
otazka_id_zarazeni	int	Identifikátor zařazení otázky (hlavní, cvičná, ukázková)
spravna	bit	Správnost odpovědi (0 - špatná, 1 - správná)
hodnoceni	bit	Hodnocení odpovědi studenta (0 - spatně, 1 - správně)
hodnoceni_text	varchar(8)	Textové hodnocení odpovědi studenta (správně, špatně)
pocet_kliku	int	Počet kliků na konkrétní odpověď
otazka_pocet_zobrazeni	int	Počet zobrazení otázky, pod kterou spadá odpověď
otazka_cas_otazka	time(0)	Čas zobrazení otázky, pod kterou spadá odpověď
otazka_cas_otazka_sec	int	Čas zobrazení otázky v sekundách, pod kterou spadá odpověď
oznacena	bit	Hodnota zda student odpověď označil (0 - ne, 1 - ano)

Tabulka 6.3 - Atributy tabulky faktů odpověď

Atribut `hodnoceni` se vyplňuje podle označení resp. neoznačení správné resp. špatně odpovědi. Atribut `pocet_kliku` je vyplněn pouze u on-line testů. Jedná se o počet označení a odznačení odpovědi.

Datový sklad systému eLogika obsahuje tyto tabulky dimenzí:

- Dimenze kategorie - `dim_kategorie`
- Dimenze kurz - `dim_kurz_info`
- Dimenze odpověď - `dim_odpoved`
- Dimenze otázka - `dim_otazka`
- Dimenze test - `dim_test`
- Dimenze třída - `dim_trida`
- Dimenze uživatel - `dim_uzivatel`

Tyto dimenzní tabulky mají shodnou strukturu jako tabulky stejných entit v databázi systému eLogika.

Speciálními tabulkami dimenzí jsou dimenze datum (`dim_date`) a dimenze čas (`dim_time`).

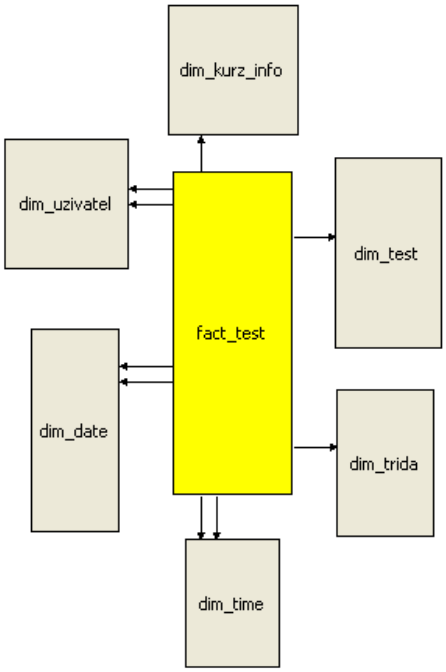
Ty jsou již přednahrány a jejich obsah se nemění.

6.3.2 Relační databázový OLAP

V druhém kroku tvorby datového skladu je potřeba zvolit technologii pro uložení dat. Jako nejvhodnější vyplynula technologie ROLAP, jejímž základem je relační model bez normalizace. Tento model využívá stávající databázové technologie v našem případě SQL Serveru 2008.

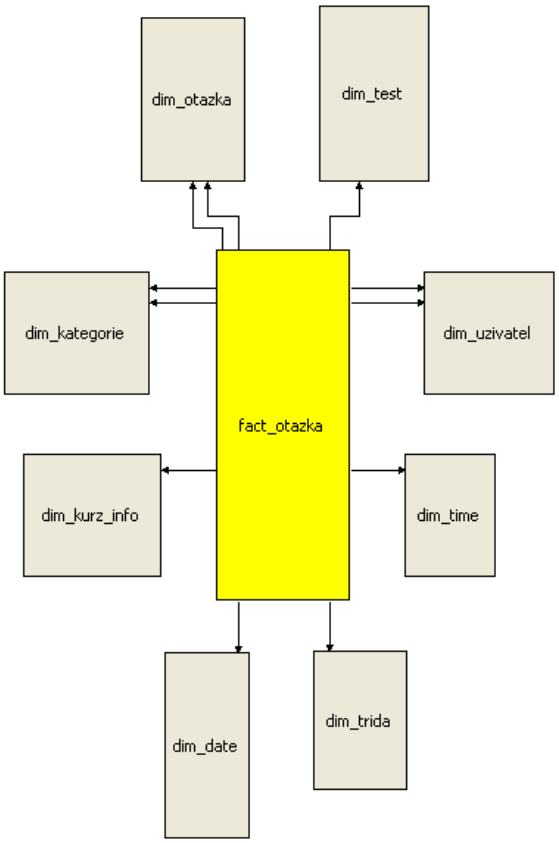
6.3.3 Hvězdicové schéma

Třetím krokem při tvorbě datového skladu je volba dimenzionálního modelu. Jedná se o topologické uspořádání (nebo schéma) tabulek faktů a dimenzí. V datovém skladu systému eLogika bylo zvoleno hvězdicové schéma. Přesněji se jedná o schéma souhvězdí, kdy některé tabulky dimenzí jsou sdíleny s více tabulkami faktů.

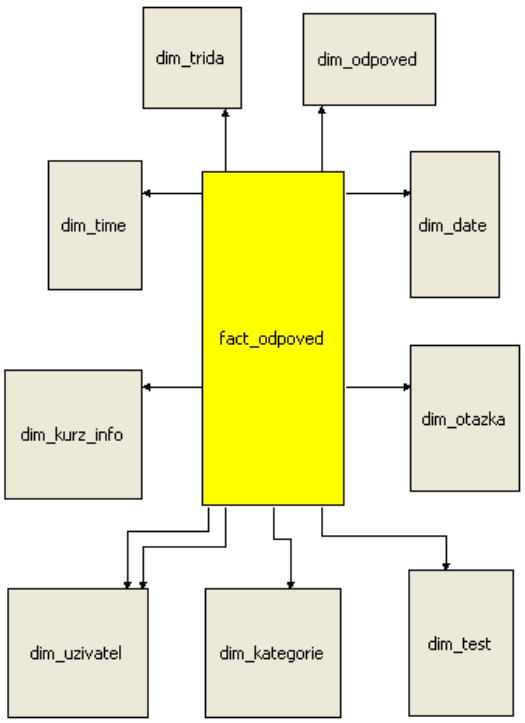


Obrázek 6.1 - Hvězdicové schéma tabulky faktů test

Na obrázku lze vidět hvězdicové schéma tabulky faktů test, která je vyznačena žlutou barvou. Šedou barvou jsou vyznačeny tabulky dimenzí. Tabulky dimenzí, které mají na tabulku faktů dvě vazby, jsou využity ve více případech. Například u tabulky dimenze „datum“ (`dim_date`) se jedná o datum vypracování testu a datum přihlášení na test.



Obrázek 6.2 - Hvězdicové schéma tabulky faktů otázka



Obrázek 6.3 - Hvězdicové schéma tabulky faktů odpověď

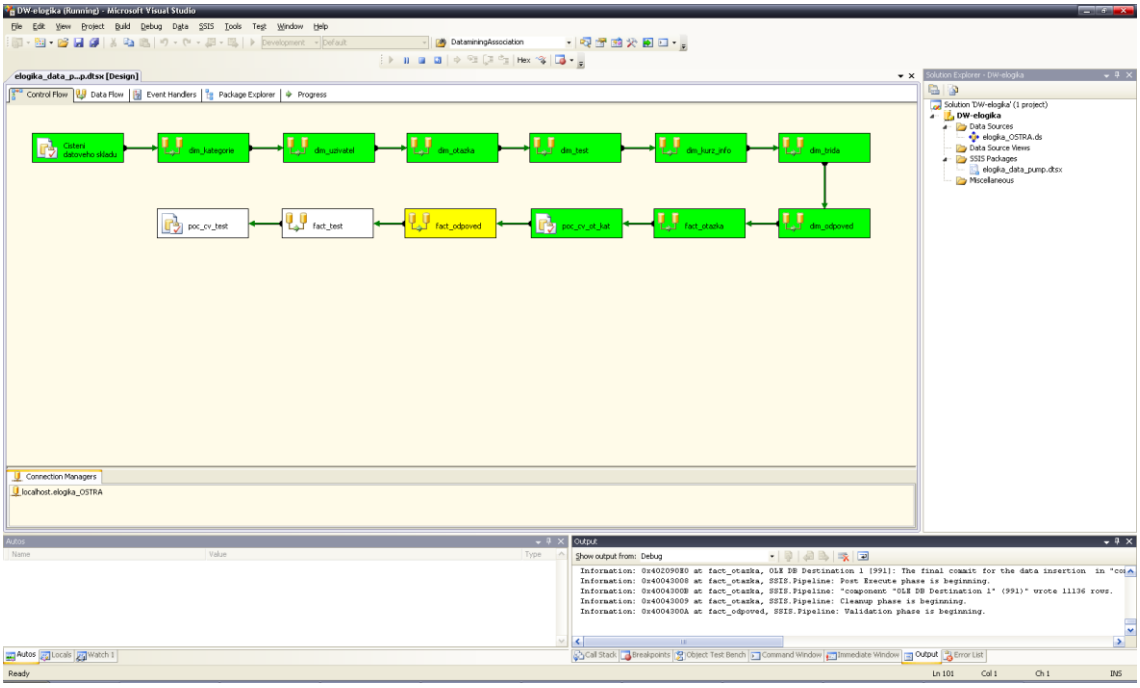
6.3.4 Datová pumpa

Závěrečným krokem je vytvoření tzv. datové pumpy, pomocí které se naplní datový sklad. Pro plnění datového skladu v systému eLogika je využito integračních služeb SQL Serveru 2008 (SQL Server 2008 Integration Services). Tato služba umožňuje správu datových toků, následnou transformaci a uložení do cílové databáze – tzv. ETL procesy. Procesy integračních služeb lze navrhnout jako jednorázovou akci, nebo akci, která se periodicky opakuje. Za jednorázovou akci můžeme považovat například migraci dat z jedné databázové platformy do druhé. Periodicky opakující se úlohy mají společnou podmínku, aby proběhly v určitém požadovaném čase.

Jednotlivé části ETL procesu:

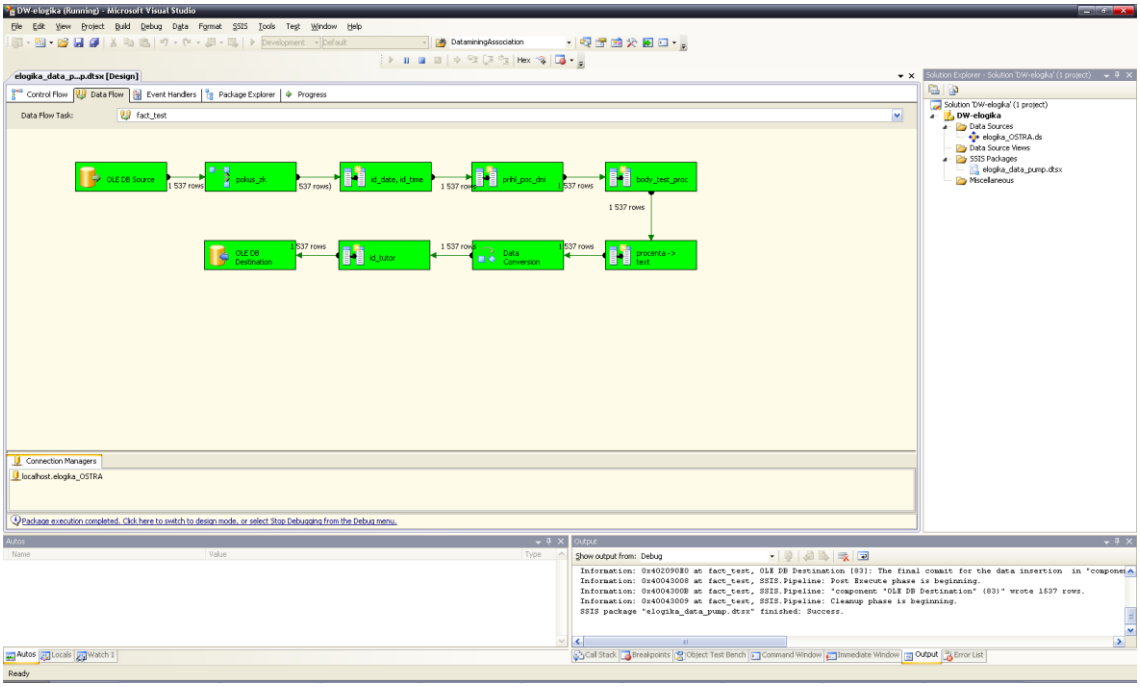
- Extract (Extrakce) – vytažení dat z určitých zdrojů
- Transform (Transformace) – ověření, úprava, čištění, integrování dat
- Load (Nahrání) – nahrání dat do datového skladu

Pro vizuální modelování, návrh, vytváření, testování a ladění projektů pro vytváření ETL procesů slouží grafický nástroj DTS Designer, který je součástí Business Intelligence Development Studia. Toto prostředí je integrováno do aplikace Microsoft Visual Studio.



Obrázek 6.4 - Datové toky v SQL Server 2008 Integration Services

Na obrázku jsou zobrazeny jednotlivé datové toky, pomoci kterých bude naplněn datový sklad systému eLogika. Každý datový tok se skládá z libovolného množství bloků pro načítání dat, transformaci dat a uložení do cílové databáze. V našem případě do tabulek datového skladu. V následujícím obrázku je ukázka jednoho z datových toků.



Obrázek 6.5 - Datové transformace v SQL Server 2008 Integration Services

Výstupem z integračních služeb SQL Serveru 2008 je tzv. SSIS balíček (SSIS Package). Jedná se o soubor s příponou *.dtsx, který obsahuje vytvořené datové toky. Tento soubor lze považovat za datovou pumpu systému eLogika. Po spuštění tohoto SSIS balíčku a provedení všech datových toků (ETL procesů) je datový sklad systému eLogika naplněn.

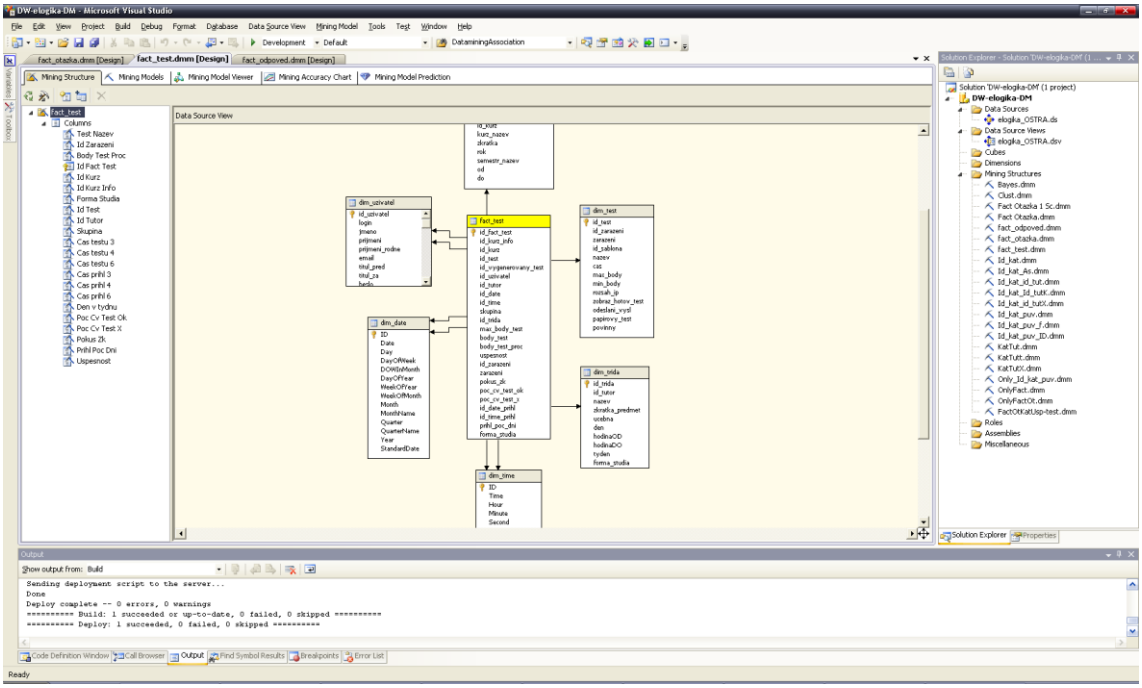
Na řadě je další fáze metodologie CRISP-DM, a to modelování data miningových metod na námi připravená data v datovém skladě.

6.4 Modelování data miningových metod

V podkapitole si popíšeme tvorbu dolovacích struktur a dolovacích modelů, které byly následně aplikovány na datový sklad systému eLogika. Funkce obou těchto prvků jsou popsány v kapitole „SQL Server 2008 a data mining“.

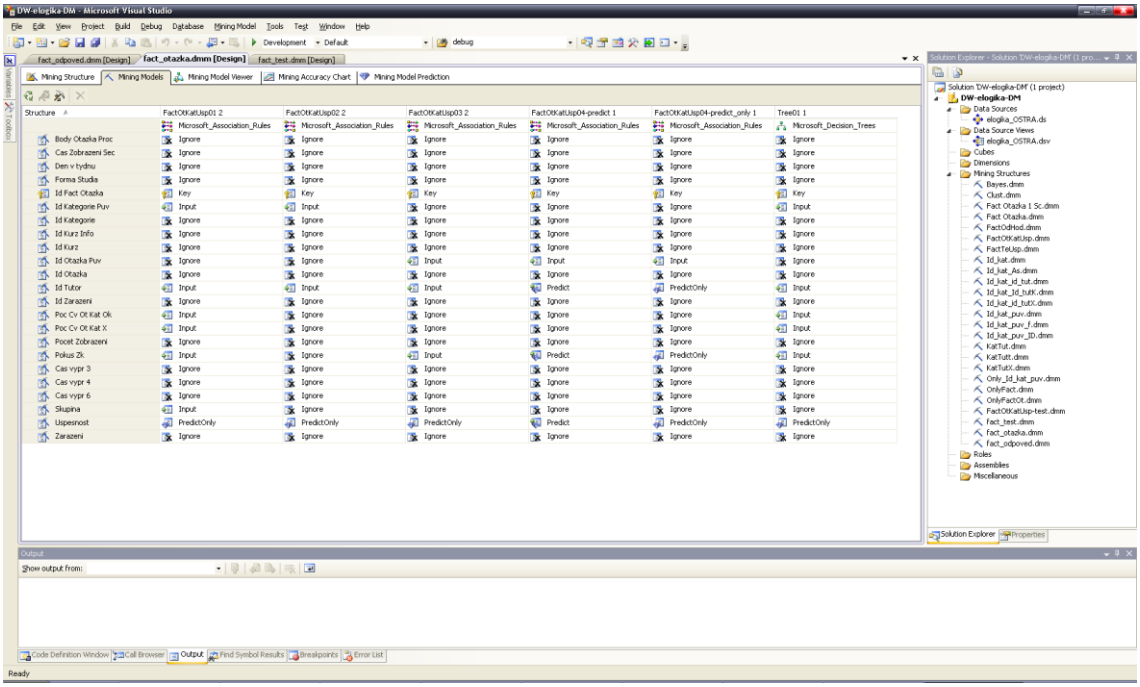
Další možností prostředí Business Intelligence Development Studio je tvorba projektu Analytických služeb (Analysis Services Project), ve kterém lze pomocí vizuálního prostředí vytvářet dolovací struktury a dolovací modely. Nejprve je potřeba definovat zdrojové zdroje a pohledy, které budou sloužit jako vstupy pro jednotlivé struktury. V našem případě je tímto zdrojem vytvořený datový sklad systému eLogika.

Po dokončení fáze definování zdrojových dat lze přistoupit k tvorbě dolovacích struktur a modelů. Pro tyto účely existuje v Business Intelligence Development Studiu průvodce Data Mining Wizard. Tato funkce uživatele provede ke zdárnému vytvoření data miningové struktury a zároveň dolovacího modelu. Strukturu lze následně upravovat a případně v ní vytvářet další dolovací modely. V projektu analytických služeb, který je vytvořený pro systém eLogika, má každá příslušná tabulka faktů vytvořenou vlastní dolovací strukturu. Jak už bylo vysvětleno výše, každá data miningová struktura může obsahovat více dolovacích modelů s použitím různých data miningových algoritmů, které využívají pouze některých atributů této struktury.



Obrázek 6.6 - Prostředí pro tvorbu dolovacích struktur

Na obrázku lze vidět prostředí Business Intelligence Development Studio, ve kterém je vytvořena dolovací struktura *fact_test*, která má jako datový zdroj nastavenu tabulku faktů `fact_test` a její tabulky dimenzí. V levé části lze vidět seznam atributů, které struktura využívá pro analýzu. Krom atributů uložených přímo v tabulce faktů lze využít i atributy dimenzí. Pomocí průniku jednotlivých dimenzí lze získat požadované hodnoty. Problém nastává u tabulek dimenzí, které mají na tabulku faktů dvě vazby. Zde je potřeba určit, která vazba se pro daný atribut využívá pro průnik dimenzí, jinak dochází k selhání dolovacího modelu. Obdobně se postupuje i u ostatních dolovacích modelů *fact_otazka* a *fact_odpoved*.



Obrázek 6.7 - Prostředí pro správu dolovacích modelů

Následující fází je vytváření dolovacích modelů pro analýzu dalších scénářů za použití různých data miningových algoritmů. Tvorbu dolovacích modelů pro systém eLogika lze popsat ve třech krocích.

6.4.1 Výběr algoritmů

Prvním krokem je výběr data miningového algoritmu. Podrobně byly tyto algoritmy popsány v kapitole „SQL Server 2008 a data mining“. Je potřeba znát požadované vstupní a výstupní atributy pro jednotlivé dolovací metody. Postupně byly v rámci vývoje vyzkoušeny všechny algoritmy obsažené v SQL Serveru 2008. Některé se ukázaly zcela nevhodnými z důvodu omezeného vstupního souboru. Například algoritmus časových řad očekává na vstupu určitou časovou posloupnost. Záznamy o provedených testech sice obsahují datum a čas provedení, ale vzhledem k tomu, že testů se provádí pouze několik za semestr, je tato časová řada velmi řídká. S ohledem k povaze vstupních dat se jako nejvhodnější ukázalo použití metod Asociačních pravidel a Rozhodovacích stromů. Obě tyto metody jsou zaměřeny na diskrétní data. Konkrétní využití těchto algoritmů bude popsáno v podkapitole „Implementace do systému eLogika“.

6.4.2 Výběr atributů

Dalším krokem je výběr správných atributů pro analýzu. V předcházejících kapitolách jsme již uvedli, že v jedné dolovací struktuře lze využít pouze vybrané atributy pro určitou situaci. Ostatní atributy budou ignorovány a do výsledků se nikterak neprojeví. Jak již bylo řečeno v předcházející části, použité data miningové metody jsou zaměřeny na diskrétní data. Pokud již je atribut kategorizován, stačí jej pouze vybrat. V opačném případě lze využít automatického diskterizování SQL Serveru 2008, který pak sám zváží rozložení hodnot a určí ideální kategorie. Obecně se pro analýzu používají názvy jednotlivých entit, například název filmu, název produktu atd. Pro testovací účely lze i v tomto případě použít například název testu nebo název

kategorie, ale pro následné využití výsledků v systému eLogika je potřeba pracovat s identifikátorem jednotlivých entit. Podrobněji se budeme tomuto tématu věnovat v podkapitole „Implementace do systému eLogika“.

6.4.3 Filtrování vstupních dat

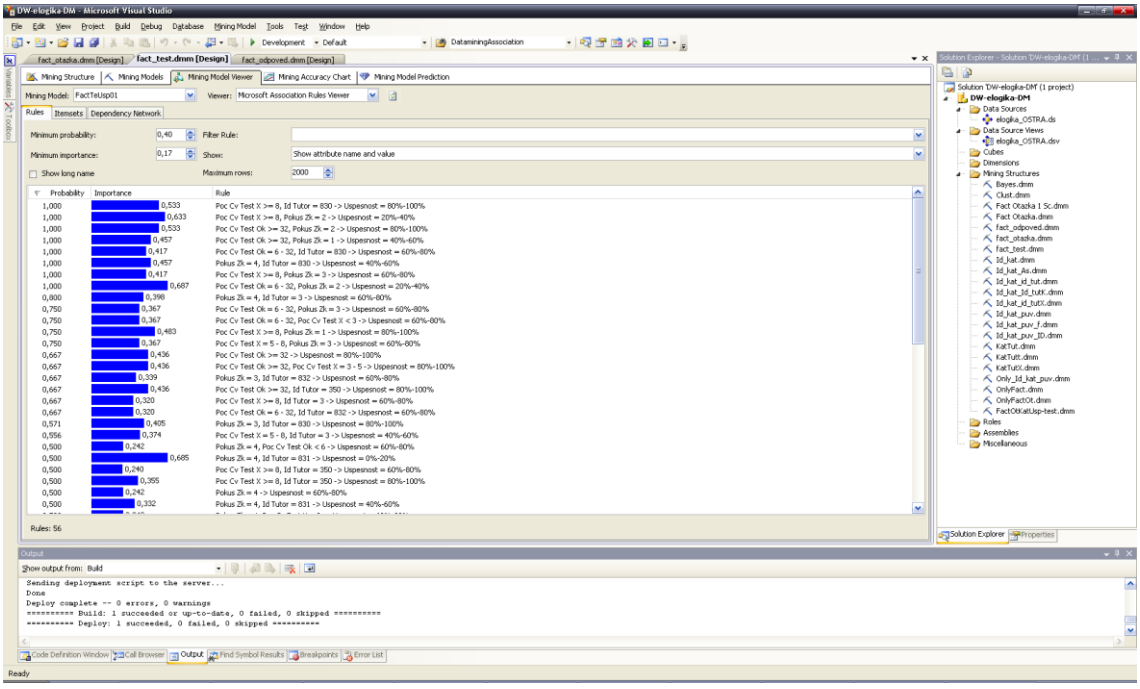
Posledním krokem je výběr vstupních dat pro jednotlivé dolovací modely. Vzhledem k tomu, že tabulky faktů obsahují informace o veškerých vypracovaných testech za kurzy vedené v systému eLogika, je potřeba tato data pro určitou situaci odfiltrovat. Nelze míchat záznamy kupříkladu z kurzu „Matematická logika“ a kurzu „Úvod do teoretické informatiky“. V každém vytvořeném dolovacím modelu lze nastavit filtr vstupních dat. Pro správné použití je potřeba znát strukturu kurzů v systému eLogika. Z uživatelského hlediska bude filtrování vstupních dat popsáno v podkapitole „Implementace do systému eLogika“.

Nyní lze takto vytvořené dolovací modely zpracovat. Pokud vše proběhne v pořádku, jsou výsledky dostupné v tzv. Prohlížeči dolovacího modelu (Mining model viewer). Vzhledem k tomu, že některé data miningové metody mají různé požadavky na vstupní data, je zapotřebí neustále iterovat mezi fází přípravy dat a modelování.

Po vytvoření dolovacích modelů pro všechny možné scénáře je dalším krokem metodologie CRISP-DM vyhodnocení výsledků.

6.5 Vyhodnocení výsledků

V této fázi je zapotřebí vyhodnotit dosažených výsledků pomocí vytvořených dolovacích modelů vzhledem ke stanoveným cílům. Jak již bylo řečeno v předcházející podkapitole, výsledky jsou k vidění v prohlížeči dolovacího modelu.



Obrázek 6.8 - Výsledky dolovacího modelu

Jak lze vidět na obrázku, výstupem dolovacího modelu je poměrně velké množství potenciálně užitečných informací. Těchto modelů byla vytvořená celá řada, proto lze očekávat, že i množství těchto potenciálně důležitých informací bude pro vyučujícího a následně i studenta dostačující. Možnosti využití těchto informací bude popsáno v závěru následující podkapitoly.

6.6 Implementace do systému eLogika

Poslední fází metodologie CRISP-DM je implementace. Jejím cílem je interpretovat získané znalosti uživateli tak, aby s nimi mohl pracovat a konkrétně je využívat. V této podkapitole bude popsáno, jakým způsobem byla provedena implementace data miningových metod do systému eLogika. Způsob prezentování výsledků a práce s těmito znalostmi.

Systém eLogika je webová aplikace vyvíjena v technologii ASP.NET, která je součástí .NET Frameworku. Tato technologie je založena na CLR, které přináší možnost implementovat projekt v jakémkoli jazyce podporovaném .NET Framework. V případě systému eLogika byl zvolen objektově orientovaný jazyk C#. Jako databázový server byl zvolen SQL Server 2008 Enterprise Edition, jehož součástí jsou i analytické služby pro práci s data miningovými metodami. Vývojovým prostředím bylo zvoleno Microsoft Visual Studio 2008, včetně jeho nástavby Business Intelligence Development Studio. Data mining v systému eLogika je vyvíjen jako externí systém, který využívá rozhraní systému eLogika, například pro získávání informací o kategoriích, otázkách atd.

6.6.1 Externí systém pro data mining

Externí systém se využívá pro práci s SQL Serverem 2008, přesněji jeho součástí Analytických služeb (Analysis services). Aby bylo možné s touto službou pracovat, je potřeba se připojit pomocí přihlašovacího řetězce tzv. ConnectionStringu (13).

```
<add name="DMConnectionString" connectionString="Data Source=NTBVP;
Initial Catalog=DW-eLogika-DM;Integrated Security=SSPI;"
providerName="System.Data.SqlClient" />
```

V ukázce je zobrazen connectionString pro připojení k analytickým službám SQL Serveru 2008.

Dalším krokem pro použití data miningových metod je potřeba naplnění datového skladu. Jak již bylo popsáno v předešlé podkapitole „Příprava dat a tvorba datového skladu“, pro naplnění datového skladu se využívá tzv. SSIS balíčků. Po provedení tohoto balíčku, resp. provedení všech ETL procesů, je datový sklad naplněn. Spuštění tohoto balíčku lze provést z uživatelského rozhraní systému eLogika.

```
Application app = new Application();
Package package = null;

try
{
      package = app.LoadPackage(@"elogika_data_pump.dtsx", null);
      Microsoft.SqlServer.Dts.Runtime.DTSExecResult results =
      package.Execute();
```

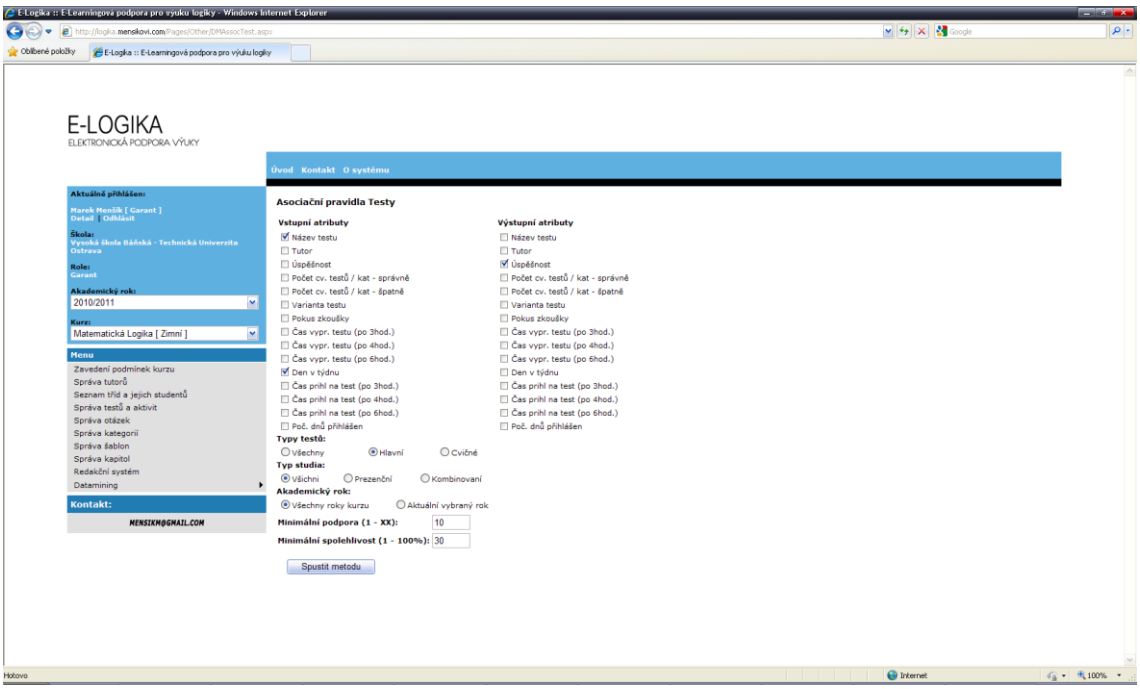
```
}
catch (Exception ex)
{
      throw ex;
}
finally
{
      package.Dispose();
      package = null;
}
```

V příkladu lze vidět spuštění *.dtsx souboru (SSIS balíčku), který naplní datový sklad systému eLogika. Tuto možnost má v rámci systému eLogika právo spustit pouze role tutor.

Hlavním požadavkem implementace bylo, aby data miningové metody byly použitelné ve všech kurzech systému eLogika, i v budoucnu vytvořených. Z tohoto důvodu bylo jedinou možností vytvářet jednotlivé dolovací modely dynamicky pro určitý kurz. Dolovací struktury pro tento účel již byly založeny ve fázi modelování, popsané výše. Do těchto struktur budou tvořeny dolovací modely. K analýze datového skladu byly implementovány tyto metody:

6.6.2 Asociační pravidla

Tento algoritmus byl zvolen z důvodu nejlepších výsledků nad vstupními daty. Je to dáno z toho důvodu, že povaha vstupních dat je převážně diskrétního typu, který je vhodný pro tuto metodu. Nyní si popíšeme, jakým způsobem je vytvářen dolovací model pro využití asociačních pravidel.



Obrázek 6.9 - Systém eLogika - Volba atributů pro metodu Asociační pravidla

Uživatel (vyučující) pomocí webového rozhraní (na obrázku) vybere vstupní a výstupní atributy,

u kterých chce najít potenciální asociace. Dále lze různým způsobem filtrovat vstupní data. Například, zda budou analyzovány všechny provedené testy nebo pouze hlavní, popř. cvičné. Další možností filtrování je typ studia, a to prezenční studenti, kombinovaní nebo všichni. Poslední možností je volba akademického roku kurzu, tato funkce bude podrobně vysvětlena v další části. V poslední řadě je třeba nastavit minimální podporu a minimální spolehlivost. Tyto charakteristiky metody asociačních pravidel jsou vysvětleny v podkapitole, která se této metodě věnuje.

Jakmile uživatel odešle požadavek na spuštění metody, data miningový systém vygeneruje dotaz v jazyku DMX, pomocí kterého bude vytvořena požadovaný dolovací model. Dále také vygeneruje DMX dotaz pro spuštění dolovacího modelu, načtení výsledků a smazání dolovacího modelu.

```
ALTER MINING STRUCTURE fact_test
ADD MINING MODEL fact_test16
(
      [Id Fact Test],
      [Id Tutor]
      [Test Nazev],
      [Poc Cv Test Ok],
      [Poc Cv Test X],
      [Den v tydnu],
      [Prihl Poc Dni],
      [Uspesnost] PREDICT_ONLY
) USING Microsoft_Association_Rules(MINIMUM_SUPPORT=10,
MINIMUM_PROBABILITY=0.30)

WITH FILTER([Id Kurz] = '16' AND [ID Zarazeni] = '1')
```

Ukázka vygenerovaného dotazu v jazyku DMX, který vytváří nový dolovací model s názvem `fact_test16` v dolovací struktuře `fact_test`. Tento název modelu je složen z názvu struktury a ID kurzu, pokud by byl ve filtru nastaven i akademický rok, je k tomuto názvu ještě doplněno ID roku. Klíčem je v tomto případě identifikátor tabulky faktů `Id Fact Test`. Lze zde vidět i použití filtru pro určitý kurz a zařazení testu (hlavní, cvičné).

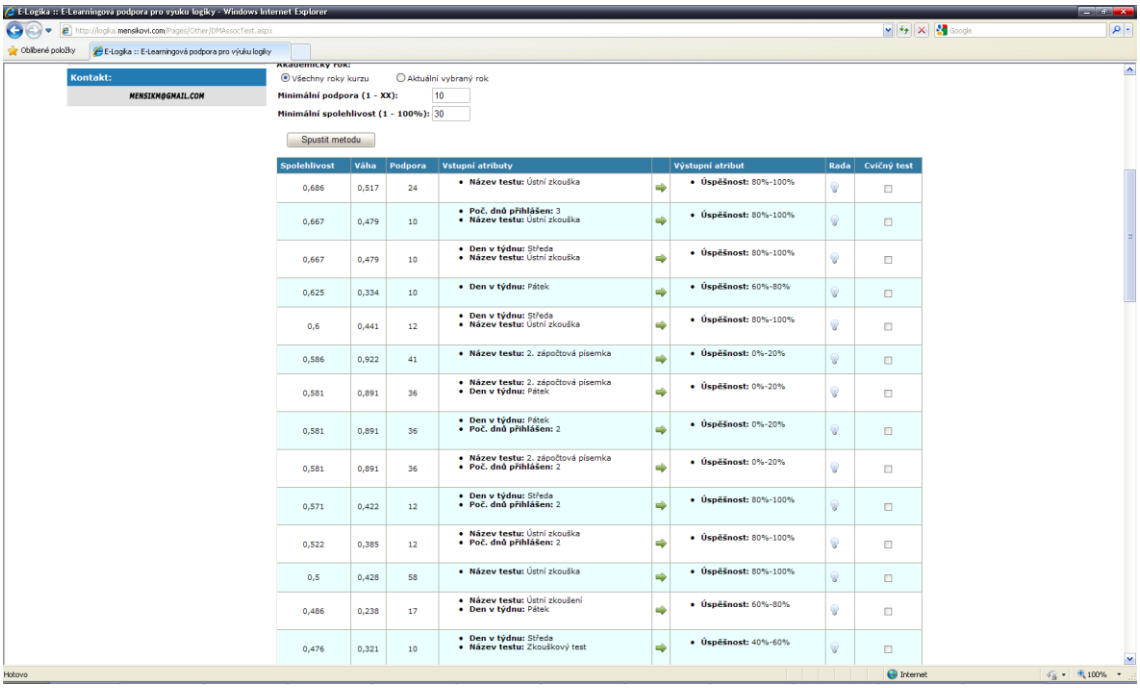
```
INSERT INTO fact_test16
```

Tímto dotazem jazyka DMX se provede spuštění dolovacího modelu `fact_test16`.

```
CALL
System.Microsoft.AnalysisServices.System.DataMining.AssociationRules.G
etRules('[fact_test16]', 0, 100, 1, 0.30, 0.0, '', True)
```

Pomocí tohoto dotazu lze z SQL Serveru 2008 načíst výsledky analýzy po provedení dolovacího modelu. Jak již bylo zmíněno výše, při analýze některých entit se používají místo jejich názvů identifikátory. Jedná se o entity Kategorie, Otázka, Odpověď a Tutor. Po provedení dolovacího modelu a načtení výsledků jsou tyto entity podle identifikátoru načteny z tabulky dimenzí. Tento

postup se provádí z důvodu následného využití výsledků data miningu, např. pro generování cvičných testů. Například pokud bychom použili pouze název kategorie, mohl by vzniknout problém při vytváření cvičného testu podle tohoto názvu, jestliže by se jiná kategorie jmenovala stejně.

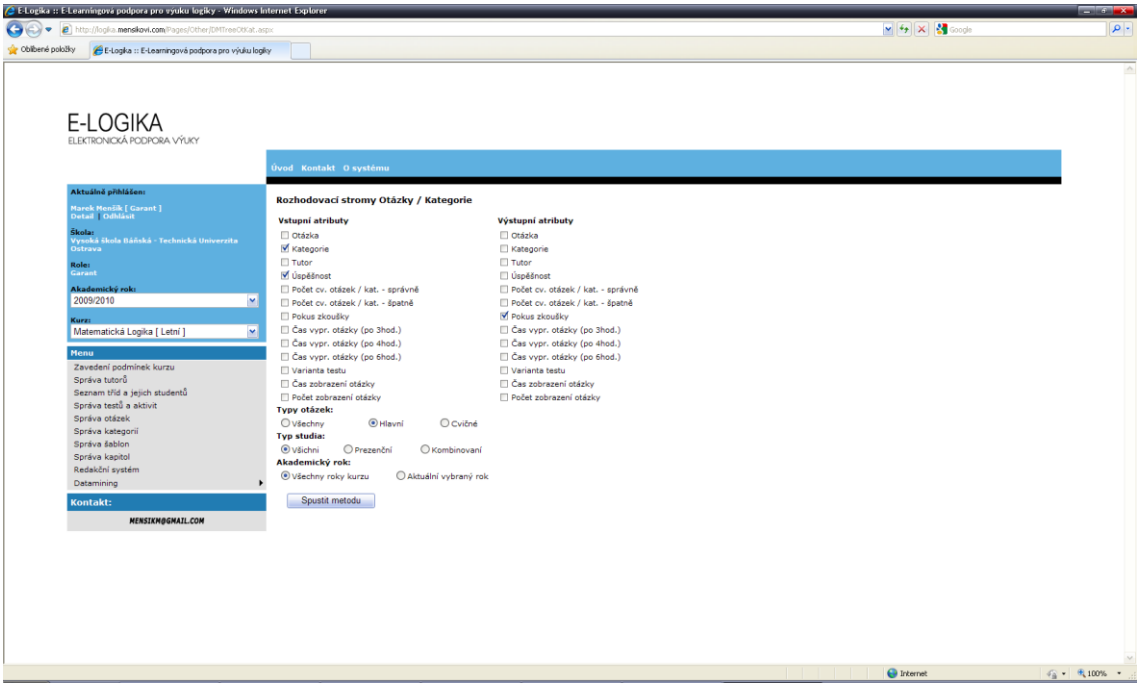


Obrázek 6.10 - Systém eLogika - Výsledky metody Asociačních pravidel

Po spuštění dolovacího modelu se výsledky interpretují uživateli do srozumitelné formy. V ukázce lze vidět výstupy metody Asociačních pravidel. Uživatel následně může vytvářet cvičné testy na problematické látce nebo studentům předat na základě výsledků užitečné rady a doporučení, tyto možnosti budou popsány níže.

6.6.3 Rozhodovací stromy

Využití algoritmů rozhodovacích stromů probíhá obdobným způsobem jako u předcházející metody Asociačních pravidel. Uživatel vybere vstupní a výstupní atributy a nastaví možnosti filtrování vstupních dat.

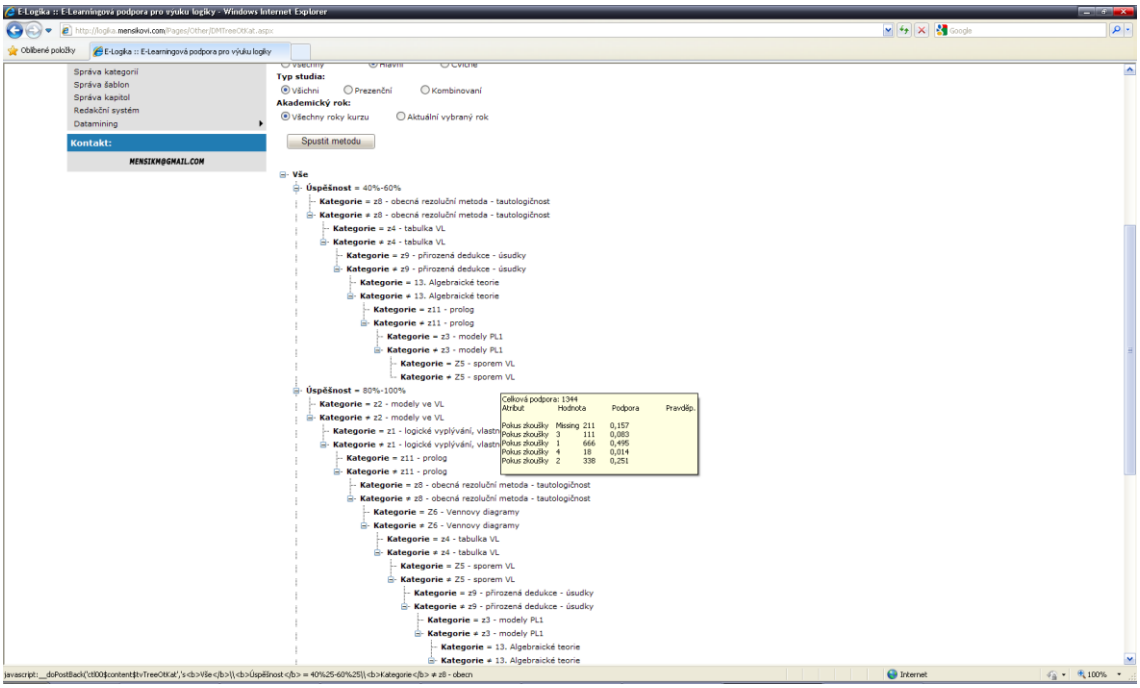


Obrázek 6.11 - Systém eLogika - Volba atributů pro metodu Rozhodovací stromy

Po následném spuštění metody se opět vygenerují DMX dotazy, které vytvářejí, spouští, načítají výsledky a mažou dolovací model.

```
SELECT FLATTENED * FROM [fact_otazka-Tree16].CONTENT
```

Ukázka DMX dotazu pro načtení výsledků z dolovacího modelu. Výsledky jsou poté uživateli interpretovány pomocí webového rozhraní.



Obrázek 6.12 - Systém eLogika - Výsledky metody Rozhodovací stromy

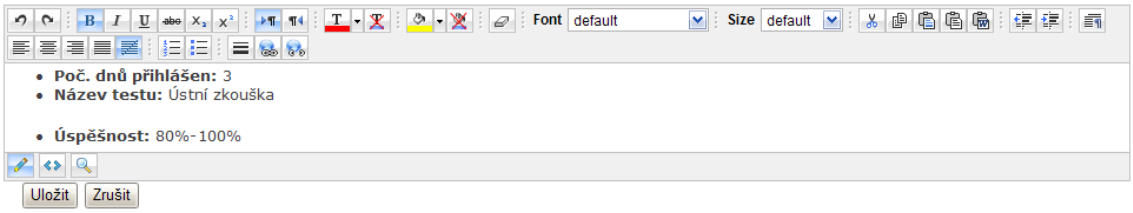
6.6.4 Historie kurzů

Z důvodu ročního (popř. pololetního) opakování jednotlivých kurzů bylo vhodné zavést propojení výsledků testů těchto kurzů. Toto propojení je prováděno pomocí tzv. navazujících tabulek, ve kterých se ukládá historická návaznost jednotlivých entit. Následně při plnění datového skladu se tyto informace zavedou do tabulky faktů a při analýze je možné pouštět data miningové metody buďto na aktuální rok kurzu, nebo na celou historii tohoto kurzu. Druhým důvodem je i postupné zvětšování vstupního analyzovaného souboru. Obecně se pro získání lepších výsledků doporučuje aplikovat data miningové metody na vstupní soubor v rozsahu desetitisíc záznamů a více. Počet záznamů o provedených testech v datovém skladě systému eLogika se odvíjí od počtu studentů jednotlivých kurzů. Každou iterací kurzu lze získat další a další záznamy o vypracovaných testech.

6.6.5 Vytváření cvičných testů a přidání rad a doporučení

Po provedení data miningových algoritmů a zobrazení výsledků má vyučující možnost využít těchto potenciálně důležitých informací pro zlepšení výuky. První možností je vytvoření cvičných testů na základě výsledků. Vyučující vybere kapitoly, popř. otázky, ve kterých mají studenti problémy, a na základě nich může vytvořit cvičný test. Tato volba je přístupná pouze, pokud výsledky dolovacího modelu obsahují jednu z těchto entit. Student tak dostává možnost procvičit si problematickou látku a vyvarovat se tak problémům.

Druhou možností je z výsledků vytvořit radu nebo doporučení pro studenta. Například z analýzy testů vyplynulo, že studenti, kteří byli na zkoušku přihlášeni 3 dny, měli u ústního zkoušení úspěšnost mezi 80%-100%. Tuto informaci může vyučující studentovi bez okolků předat. Vyvstává zde ale problém se sdělitelností určitých výsledků. Pokud jsou například prováděny analýzy nad otázkami, není vhodné, aby zadání otázky určené pro zkoušení bylo vkládáno do těchto rad a doporučení.



Obrázek 6.13 - Systém eLogika - Vytvoření rady a doporučení

Na obrázku lze vidět WYSIWYG editor pro vkládání rad a doporučení studentovi. Tyto rady lze následně editovat a upravovat. Je nutné podotknout, že výsledky data miningových metod musí vyhodnotit uživatel znalý struktury kurzu a vygenerovaných testů.

7 Závěr

V dnešní době se stále více využívá moderních technologií v různých oborech. Jedním z těchto oborů je bezesporu vzdělávání. Dostupnost počítačů sebou přinesla další možnosti vzdělávacích metod, jako je například e-learning. Této metody využívá i systém eLogika, který je určen pro vyučující a také studenty. Každý takovýto systém si za určitou dobu provozu vytvoří rozsáhlou databázi. Ta v sobě může skrývat nesmírné informační bohatství. Problémem však je, jak ho z uložených dat odkrýt. Zde nastupuje na řadu poměrně mladý obor informatiky - data mining. Pomocí data miningových algoritmů lze takovéto rozsáhlé databáze analyzovat a získávat z nich potenciálně důležité a užitečné informace.

Cílem této práce bylo využít data miningu k analýze dat získaných systémem eLogika. Systém eLogika je již druhým rokem využíván na Vysoké škole báňské – Technické univerzitě Ostrava převážně pro výuku Matematické logiky, ale i jiných kurzů. Za tuto dobu testovacího provozu systému eLogika proběhly dva semestry kurzu Matematické logiky. Evidované informace o studentech, například provedených hlavních testech, cvičných testech, vyučujících atd. jsou vstupními daty pro následnou analýzu pomocí data miningových metod. Nejprve bylo potřeba z těchto informací vytvořit datový sklad, který byl poté naplněn. Tato data byla následně analyzována pomocí vybraných data miningových metod. Výsledky byly zpracovány a interpretovány uživateli ve formě, která umožňuje další cílený rozvoj při studiu a výuce.

Hlavním přínosem diplomové práce je snaha pomoci studentovi v jeho cestě absolvování jednotlivých kurzů. Předat studentovi informace, které mu co nejvíce ulehčí studium, nebo pomohou procvičit problematickou látku. To je prováděno pomocí tzv. cvičných testů vygenerovaných na základě výsledků z data miningových metod. Další možností je vytvoření užitečných rad a doporučení studentovi. Vyučující na základě výsledků z dolování dat vyhodnotí pro studenta užitečné informace a ty mu může v přehledné formě předat. Výhodou je také možnost využití data miningových metod i v budoucích kurzech vedených v systému eLogika.

Do budoucna by bylo ještě možné rozšířit množství vstupních dat o studentovi pro následnou analýzu a získání lepších výsledků z dolovacích metod. Například použití různých forem prezentace vyučované látky a sledování vlivu na studentovu připravenost u testů.

8 Použitá literatura

1. **Wikipedie, Přispěvatelé.** Learning Management System. *Wikipedie: Otevřená encyklopedie.* [Online] 2. Únor 2011. http://cs.wikipedia.org/wiki/Learning_Management_System.

2. —. E-learning. *Wikipedie: Otevřená encyklopedie.* [Online] 21. Březen 2011. http://cs.wikipedia.org/wiki/E-learning.

3. **Berka, Petr.** *Dobývání znalostí z databází.* Praha : Academia, 2003. 80-200-1062-9.

4. **Ota Novotný, Jan Pour, David Slánský.** *Business intelligence : jak využít bohatství ve vašich datech.* Praha : Grada, 2005. 80-247-1094-3.

5. **Rud, Olivia Parr.** *Data mining : praktický průvodce dolováním dat pro efektivní prodej, cílený marketing a podporu zákazníků (CRM) [překlad Ivo Magera, Milan Daněk].* Praha : Computer Press, 2001. 80-7226-577-6.

6. **Lacko, Luboslav.** *Business Inteligence v SQL Serveru 2008.* místo neznámé : Computer Press, a.s., 2009. 978-80-2512887-9.

7. **Jamie MacLennan, ZhaoHui Tang, Bogdan Crivat.** *Data Mining with Microsoft SQL Server 2008.* Indiana : Wiley Publishing, Inc., 2009. 978-0-470-27774-4.

8. **Larson, Brian.** *Delivering Business Intelligence with Microsoft SQL Server 2008.* místo neznámé : The McGraw-Hill Companies, 2009. 978-0-07-154945-5.

9. **Poulíček, Jakub.** *Data mining a SQL Server 2008.* [Bakalářská práce] Ostrava : VŠB-TU Ostrava, FEI, 2010.

10. **Hotek, Mike.** *Microsoft SQL Server 2008 - Krok za krokem.* místo neznámé : Computer Press, 2009. 978-80-251-2466-6.

11. **Rakesh Agrawal, Tomasz Imielinski, Arun Swami.** Mining Association Rules between Sets of Items in Large Databases. [Online] 1993. [Citace: 2. duben 2011.] http://rakesh.agrawal-family.com/papers/sigmod93assoc.pdf.

12. **Walters, Robert E.** *Mistrovství v Microsoft SQL Server 2008 : [kompletní průvodce databázového experta].* [překl.] Pavel Poloncý. Brno : Computer Press, 2009. str. 864. 978-80-251-2329-4 (váz.).

13. Connection strings for SQL Server 2008. *ConnectionStrings.com.* [Online] http://www.connectionstrings.com/sql-server-2008.

Přílohy

A. Obsah CD:

- DB – Skripty pro obnovení databází
- Doc – Dokumentace
- elogika_mservice – Projekt systému eLogika
- elogika_ds – Webový projekt systému eLogika
- DW-elogika – SSIS Projekt
- DW-elogika-DM – Analysis Services Projekt
- Instalace.docx – Postup spuštění systému eLogika